Table of Contents

- Introduction
- What's New in this Release
- Version History
- MvcDiagram Fundamentals
- Getting Started
  - Add Nuget Package
  - Create DiagramView in MVC Project
  - Create DiagramView in Razor Pages Project
- User Interaction
- Programming Interface Overview
  - Model-View Architecture
  - User Interface Components
  - Diagram Structure
    - Standard Item Types
    - Custom Composite Nodes
    - Creating and Deleting Items
    - Accessing Items
    - Attaching and Grouping Items
  - Laying Out Diagrams Automatically
  - Diagram Serialization and Output
    - Serializing the Diagram
    - Exporting the Diagram
  - Appearance
    - Styles and Themes
    - Brush and Pen Classes
    - Node Effects
  - Client Side API Overview
    - Diagram Structure
      - Standard Item Types
      - Composite Nodes
      - Creating and Deleting Items
      - Accessing Items
      - Collapsing and Expanding Tree Branches
      - Attaching and Grouping Items
    - User Interaction
      - Diagram Events
      - Controlling User Interaction
      - Alignment Grid
      - Undo and Redo
      - Scrolling and Zooming
      - Testability
    - Placement and Layout
      - Items Placement and Z-order
      - Lane Grid
      - Anchor Points and Patterns
      - Link Routing
      - Laying out the Diagram
    - Appearance
      - Customizing Node Appearance
      - Node Effects
      - Customizing Link Appearance
      - Styles and Themes
      - CSS Styling
      - Define Shape Geometries
    - Animations
  - .NET Core Assemblies
  - Initial Values of Item Properties
- Tutorials
  - Tutorial 1: Getting started
  - Tutorial 2: Handling client-side events
  - Tutorial 3: Overview and NodeListView
  - Tutorial 4: Loading graph data
  - Tutorial 5: Loading hierarhical data
  - Tutorial 6: Create a Custom Node Type
  - Tutorial 7: Custom Nodes Integration
  - Tutorial 8: Custom Composite Nodes
- Reference
  - Client Side Reference
    - MindFusion.Animations
      - Animation Class

- AnimationType Enumeration
- EasingType Enumeration
- Events Class

MindFusion.Collections
- Grid Class
- Queue Class

MindFusion.Controls
- Alignment Enumeration
- CancelEventArgs Class
- Canvas Class
- CanvasControl Class
- Control Class
- Disposable Class
- DomUtils Class
- EventArgs Class
- EventDispatcher Class
- Events Class
- MouseButton Enumeration
- MouseCursors Enumeration
- TickPosition Enumeration
- WebComponents Class
- ZoomControl Class

MindFusion.Diagramming
- Action Enumeration
- AddItemCommand Class
- AdjustmentHandles Enumeration
- AeroEffect Class
- Alignment Enumeration
- AnchorPattern Class
- AnchorPoint Class
- ArrowHeads Class
- AutoResize Enumeration
- Behavior Enumeration
- BehaviorBase Class
- Border Class
- Builder Class
- Button Class
- CaptionedNode Class
- Cell Class
- CellEventArgs Class
- CellFrameStyle Enumeration
- ChangeItemCommand Class
- ColumnStyle Enumeration
- Command Class
- CompatConfig Class
- ComponentEventArgs Class
- CompositeCommand Class
- CompositeNode Class
- CompositeRouter Class
- ConnectionPoint Class
- ConnectionStyle Enumeration
- ContainerNode Class
- ContainerPage Class
- ContainerPagesCommand Class
- ControlNode Class
- CreateLinkController Class
- CreateNodeController Class
- CreateSelectionController Class
- DelKeyAction Enumeration
- Diagram Class
- DiagramBase Class
- DiagramDocument Class
- DiagramEventArgs Class
- DiagramItem Class
- DiagramLink Class
- DiagramNode Class
- DiagramPage Class
- DiagramView Class
- DirectionConstraint Enumeration
- DoNothingBehavior Class
- DrawAnchorPointEventArgs Class
- DrawLinksBehavior Class
- DrawNodesBehavior Class
- DrawShapesBehavior Class
- DummyNode Class
- EffectPhase Enumeration
- Events Class

ExpandButtonAction Enumeration
Factory Class
FitSize Enumeration
FoldContainerCommand Class
FreeFormNode Class
GlassEffect Class
GlassEffectType Enumeration
GridColumn Class
GridPanel Class
GridRouter Class
GridRow Class
GridStyle Enumeration
HandlesStyle Enumeration
HandlesVisualStyle Class
HeaderEventArgs Class
HeaderResizeEventArgs Class
InplaceEditEventArgs Class
Interaction Class
InteractionState Class
ItemEventArgs Class
ItemLabel Class
ItemListView Class
LabelPosition Enumeration
LengthType Enumeration
LinkCrossingEventArgs Class
LinkCrossings Enumeration
LinkEventArgs Class
LinkLabel Class
LinkNodesBehavior Class
LinkShape Enumeration
LinkShapesBehavior Class
LinkTextStyle Enumeration
LinkValidationEventArgs Class
ListViewBase Class
MagnifyBehavior Class
Manipulator Class
MarkStyle Enumeration
ModificationStart Enumeration
ModifierKeyAction Enumeration
ModifierKeyActions Class
ModifyBehavior Class
ModifyLinkController Class
ModifyNodeController Class
ModifySelectionController Class
MouseButtonActions Enumeration
MouseInputDispatcher Class
MouseWheelAction Enumeration
MoveNodesBehavior Class
NodeConstraints Class
NodeEffect Class
NodeEventArgs Class
NodeLabel Class
NodeListView Class
Orientation Enumeration
Overview Class
PageContainerCommand Class
PagedContainerNode Class
PanBehavior Class
PanController Class
Panel Class
Path Class
PathFinder Class
PatternRouter Class
RelativeToLink Enumeration
RelativeToNode Enumeration
RemoveItemCommand Class
ResizeTableColumnCommand Class
ResizeTableRowCommand Class
RoutePattern Class
Router Class
RouteStep Class
RouteStepKind Enumeration
Ruler Class
SaveToStringFormat Enumeration
ScaleMode Enumeration
ScrollTableCommand Class
Selection Class

SelectionChangedEventArgs Class
SelectionEventArgs Class
SelectOnlyBehavior Class
SerializeTagEventArgs Class
ShadowsStyle Enumeration
Shape Class
ShapeDecoration Class
ShapeElement Class
ShapeLibrary Class
ShapeNode Class
Shapes Class
ShowAnchors Enumeration
SimplePanel Class
SimpleShape Enumeration
SinglePointerController Class
StackPanel Class
Style Class
SvgContent Class
SvgExporter Class
SvgNode Class
TableColumn Class
TableConnectionPoint Class
TableNode Class
TableRow Class
Theme Class
TreeItemEventArgs Class
TreeViewConnectionPoint Class
TreeViewItem Class
TreeViewNode Class
UndoEventArgs Class
VideoNode Class
WebComponents Class
XmlPersistContext Class
MindFusion.Diagramming.Animations
BounceAnimation Class
FillColorAnimation Class
ItemAnimation Class
OpacityAnimation Class
PathAnimation Class
PulseAnimation Class
ScrollAnimation Class
ShakeAnimation Class
StrokeColorAnimation Class
ViewAnimation Class
ZoomAnimation Class
MindFusion.Diagramming.Controls
Events Class
PageEventArgs Class
Palette Class
ShapeDesigner Class
TabAlignment Enumeration
TabbedDiagramView Class
WebComponents Class
MindFusion.Diagramming.Lanes
Cell Class
Grid Class
Header Class
ResizeType Enumeration
Style Class
MindFusion.Drawing
Alignment Enumeration
Border3D Class
Component Class
ComponentBase Class
Container Class
DashStyle Enumeration
DrawingUtils Class
Ellipse Class
Font Class
FontStyle Enumeration
GraphicsUnit Class
Image Class
ImageAlign Enumeration
LayoutAlignment Enumeration
LineJoin Enumeration
Path Class
Point Class

- Rect Class
- Size Class
- StringAlignment Enumeration
- StringFormat Class
- Text Class
- Thickness Class
- Vector Class
- Video Class
- Visibility Enumeration

MindFusion.Graphs

- Anchoring Enumeration
- AssistantPosition Enumeration
- AssistantType Enumeration
- BorderedTreeLayout Class
- Edge Class
- FlowchartLayout Class
- FractalLayout Class
- Graph Class
- GridLayout Class
- LayeredLayout Class
- Layout Class
- LayoutDirection Enumeration
- MultipleGraphsPlacement Enumeration
- Orientation Enumeration
- OrthogonalLayout Class
- Path Class
- PathFinder Class
- RadialTreeLayout Class
- SpringLayout Class
- TopologicalLayout Class
- TreeLayout Class
- TreeLayoutLinkType Enumeration
- TreeMapLayout Class
- Vertex Class

Server Side Reference

MindFusion.Diagramming

- Action Enumeration
- AdjustmentHandle Class
- AdjustmentHandles Enumeration
- AeroEffect Class
- Anchoring Enumeration
- AnchorPattern Class
- AnchorPoint Class
- AnchorPointCollection Class
- AnchorPointEventArgs Class
- ArcTemplate Class
- AreaEventArgs Class
- ArrowHeads Class
- AttachTo Enumeration
- AttachToLink Enumeration
- AttachToNode Enumeration
- AutoJson Class
- AutoResize Enumeration
- Axis Enumeration
- Behavior Enumeration
- BezierTemplate Class
- BranchIndicator Enumeration
- CellEventArgs Class
- CellFrameStyle Enumeration
- CellValidationEventArgs Class
- ClipboardEventArgs Class
- CollectionBase(T) Class
- ColumnStyle Enumeration
- CommandHistory Class
- CompositeNode Class
- CompositeNodeStyle Class
- CompositeRouter Class
- ConnectionPoint Class
- Constants Class
- ContainerChildEventArgs Class
- ContainerNode Class
- ContainerNodeProperties Class
- ContainerNodeStyle Class
- ContainerPage Class
- ContainmentKeys Enumeration
- ControlMouseAction Enumeration
- ControlPointEventArgs Class

Converter Class
CreatePathDelegate Delegate
CustomDraw Enumeration
DelKeyAction Enumeration
Diagram Class
DiagramBase Class
DiagramDocument Class
DiagramEventArgs Class
DiagramItem Class
DiagramItemCollection Class
DiagramItemProperties Class
DiagramItemState Class
DiagramItemStyle Class
DiagramLink Class
DiagramLinkCollection Class
DiagramLinkProperties Class
DiagramLinkState Class
DiagramLinkStyle Class
DiagramNode Class
DiagramNodeCollection Class
DiagramNodeEventArgs Class
DiagramNodeProperties Class
DiagramNodeState Class
DiagramNodeStyle Class
DiagramPage Class
DiagramPageEventArgs Class
DiagramPageValidationEventArgs Class
DiagramStyle Class
DirectionConstraint Enumeration
DraggedItem Class
DraggedLink Class
DraggedNode Class
DraggedShape Class
DragIndicator Enumeration
DrawAnchorPointEventArgs Class
DrawBranchIndicatorEventArgs Class
DrawCellEventArgs Class
DrawItemEventArgs Class
DrawLinkEventArgs Class
DrawNodeEventArgs Class
DummyNode Class
EditCellTextEventArgs Class
EditLinkTextEventArgs Class
EditNodeTextEventArgs Class
EffectsProvider Class
ElementTemplate Class
EmbossEffect Class
ExpandButtonAction Enumeration
ExpandButtonPosition Enumeration
Factory Class
FitSize Enumeration
FreeFormHandle Class
FreeFormNode Class
FreeFormNodeStyle Class
GlassEffect Class
GlassEffectType Enumeration
Granularity Enumeration
GridRouter Class
GridStyle Enumeration
Group Class
Group.Attachment Class
GroupAnchorStyles Enumeration
GroupCollection Class
GroupEventArgs Class
GroupRotation Enumeration
HandlesStyle Enumeration
HandlesVisualStyle Class
HitTestEventArgs Class
HitTestPriority Enumeration
HitTestResult Enumeration
HtmlBuilder Class
HyperLinkStyle Enumeration
IDiagramView Interface
IEffectsProvider Interface
ILinkRouter Interface
IMeasureUnitConverter Interface
InplaceEditable Interface

InplaceEditableList Interface
InstantiateItemEventArgs Class
InteractionState Class
IProperty Interface
IResolution Interface
ItemCollectionBase(T) Class
ItemEventArgs Class
ItemLabel Class
JsonPersistContext Class
JsonSerializeTagEventArgs Class
Key Enumeration
LabelPosition Enumeration
Layer Class
LayerEventArgs Class
LineTemplate Class
LinkAdjustmentHandle Class
LinkCrossingEventArgs Class
LinkCrossings Enumeration
LinkEventArgs Class
LinkLabel Class
LinkShape Enumeration
LinkShapeEventArgs Class
LinkTextStyle Enumeration
LinkValidationEventArgs Class
MarkStyle Enumeration
MeasureUnit Class
MeasureUnitType Enumeration
ModificationEffect Enumeration
ModificationStart Enumeration
ModifierKeyAction Enumeration
MouseButton Enumeration
MouseButtonActions Enumeration
MouseWheelAction Enumeration
NodeAdjustmentHandle Class
NodeConstraints Class
NodeEffect Class
NodeEventArgs Class
NodeHandleType Enumeration
NodeLabel Class
NodeValidationEventArgs Class
NotATreeException Class
NotifyCollectionChangedAction Enumeration
NotifyCollectionChangedEventArgs(T) Class
ObservableCollection(T) Class
Orientation Enumeration
PagedContainerNode Class
PageIconRenderer Class
Path Class
PathCollection Class
PathFinder Class
PatternRouter Class
PointCollection Class
PolylineRouting Enumeration
QuickRouter Class
ReadOnlyCollection(T) Class
ReadOnlyDiagramItemCollection Class
ReadOnlyDiagramLinkCollection Class
ReadOnlyDiagramNodeCollection Class
Relationship Enumeration
RelativeToLink Enumeration
RelativeToNode Enumeration
RenderOptions Class
RenderTarget Enumeration
RepaintEventArgs Class
RerouteLinks Enumeration
RestrictToBounds Enumeration
RoundRectangleTemplate Class
RoutePattern Class
RouteStep Class
RouteStepKind Enumeration
RoutingOptions Class
RowEventArgs Class
SaveToStringFormat Enumeration
ScriptHelper Class
Scrollable Interface
Selection Class
SelectionCopy Class

- SelectionEventArgs Class
- SelectionStyle Enumeration
- SelectionValidationEventArgs Class
- SerializeTagEventArgs Class
- ShadowsStyle Enumeration
- Shape Class
- ShapeAdjustmentHandle Class
- ShapeCollection Class
- ShapeControlPoint Class
- ShapeConverter Class
- ShapeData Class
- ShapeDecoration Class
- ShapeEventArgs Class
- ShapeFormulaParam Class
- ShapeLibrary Class
- ShapeNode Class
- ShapeNodeProperties Class
- ShapeNodeState Class
- ShapeNodeStyle Class
- Shapes Class
- ShowAnchors Enumeration
- SimpleShape Enumeration
- SmoothShadowEffect Class
- SnapToAnchor Enumeration
- Style Class
- StylePropertyChangedEventArgs Class
- SvgNode Class
- SvgNodeProperties Class
- SvgNodeStyle Class
- TableConnectionStyle Enumeration
- TableNode Class
- TableNode.Cell Class
- TableNode.CellCollection Class
- TableNode.Column Class
- TableNode.ColumnCollection Class
- TableNode.Row Class
- TableNode.RowCollection Class
- TableNodeProperties Class
- TableNodeStyle Class
- TableScrollUI Enumeration
- Theme Class
- Thickness Structure
- ThicknessConverter Class
- TimerEventArgs Class
- TreeEnumerator Class
- TreeItemEventArgs Class
- TreeItemValidationEventArgs Class
- TreeViewConnectionPoint Class
- TreeViewConnectionStyle Enumeration
- TreeViewItem Class
- TreeViewNode Class
- TreeViewNodeStyle Class
- UndoEventArgs Class
- UndoManager Class
- UndoValidationEventArgs Class
- UnitType Enumeration
- Vector Structure
- VideoNode Class
- View3D Enumeration
- VisitNode Delegate
- XmlOptions Class
- XmlPersistContext Class

MindFusion.Diagramming.Export

- AutoScale Enumeration
- DxfExporter Class
- ExcelExporter Class
- ExportLanes Enumeration
- PdfExporter Class
- SvgExporter Class
- Visio2013Exporter Class
- VisioExporter Class

MindFusion.Diagramming.Import

- ConnectorImportedEventArgs Class
- DrawImporter Class
- GroupImportedEventArgs Class
- ImportItemEventArgs Class
- ShapeImportedEventArgs Class

SvgImporter Class
Visio2013Importer Class
VisioImporter Class
MindFusion.Diagramming.Layout
AnnealLayout Class
CascadeLayout Class
CircularLayout Class
CompositeLayout Class
DecisionLayout Class
Direction Enumeration
FlowchartLayout Class
FractalLayout Class
GridLayout Class
HierarchicalLayout Class
ILayout Interface
ISwimlaneGrid Interface
LaneGrid Class
LaneType Enumeration
LayeredLayout Class
LayeredLayoutLinkType Enumeration
LayeredLayoutStatistics Class
Layout Class
LayoutLink Delegate
LayoutNode Delegate
LayoutProgress Delegate
LayoutSubgraph Delegate
MultipleGraphsPlacement Enumeration
OneWayLayout Class
Orientation Enumeration
OrthogonalLayout Class
OrthogonalRouter Class
SpringLayout Class
SpringLayoutLinkType Enumeration
SwimlaneLayout Class
TopologicalLayout Class
TreeLayout Class
TreeLayoutBalance Enumeration
TreeLayoutDirections Enumeration
TreeLayoutLinkType Enumeration
TreeLayoutType Enumeration
TreeMapLayout Class
TriangularLayout Class
MindFusion.Diagramming.Mvc
CompositeNodeConverter Class
ContainerNodeConverter Class
ControlNode Class
ControlNodeConverter Class
ControlNodeStyle Class
DiagramItemConverter Class
DiagramLinkConverter Class
DiagramNodeConverter Class
DiagramView Class
DiagramViewConverter Class
DiagramViewExtensions Class
DiagramViewTagHelper Class
FreeFormNodeConverter Class
ItemListView Class
ItemListViewExtensions Class
ItemListViewTagHelper Class
ItemTypeResolver Class
LaneGridConverter Class
LaneHeaderCollectionConverter Class
LaneHeaderConverter Class
LaneStyleConverter Class
NodeListView Class
NodeListViewExtensions Class
NodeListViewTagHelper Class
Orientation Enumeration
Overview Class
OverviewExtensions Class
OverviewTagHelper Class
Palette Class
PaletteExtensions Class
PaletteTagHelper Class
Ruler Class
RulerExtensions Class
RulerTagHelper Class

ScaleMode Enumeration
ShapeNodeConverter Class
SvgNodeConverter Class
TabAlignment Enumeration
TabbedDiagramView Class
TabbedDiagramViewExtensions Class
TabbedDiagramViewTagHelper Class
TableNodeConverter Class
TickPosition Enumeration
TreeViewNodeConverter Class
VideoNodeConverter Class
ZoomControl Class
ZoomControlExtensions Class
ZoomControlTagHelper Class
MindFusion.Drawing
Brush Class
Brushes Class
CardinalSpline Class
CombineMode Enumeration
ContentAlignment Enumeration
CursorHint Enumeration
DashStyle Enumeration
FillMode Enumeration
Font Class
FontManager Class
FontStyle Enumeration
GraphicsFactory Class
GraphicsType Enumeration
GraphicsUnit Enumeration
HatchBrush Class
HatchStyle Enumeration
HorizontalAlignment Enumeration
IGraphics Interface
IGraphicsPath Interface
ImageAlign Enumeration
IMatrix Interface
LinearGradientBrush Class
LineJoin Enumeration
MatrixOrder Enumeration
MauiGraphics Class
MauiMatrix Class
PathGradientBrush Class
Pen Class
PenAlignment Enumeration
PenUnit Enumeration
SolidBrush Class
StringAlignment Enumeration
StringFormat Class
TextAlignment Enumeration
TextureBrush Class
VerticalAlignment Enumeration
WrapMode Enumeration
MindFusion.Layout
AnnealLayoutStatistics Class
AssistantType Enumeration
BaseLayout Enumeration
ConnectorRules Class
FlowLayoutKeys Class
FlowLayoutRules Class
FlowLayoutTraits Enumeration
HierarchicalLayoutTraits Enumeration
LayeredLayoutTraits Enumeration
LayoutRules Class
NodeRules Class
OrderDirection Enumeration
Pair Class
PartitionMethod Enumeration
PathNode Class
Point3D Structure
PointList Class
PreferredRules Enumeration
Routable Interface
RouteDescriptor Class
RouteHeuristics Delegate
RouteHeuristics Delegate
RoutingGrid Interface
RoutingOptions Interface

SimpleKeys Class
SimpleLinkLogic Class
SimpleNodeLogic Class
SimpleOrientation Enumeration
SimpleRules Class
SpringLayoutMethod Enumeration
SpringLayoutTraits Enumeration
SwimlaneLayoutTraits Enumeration
TreeLayoutTraits Enumeration
TreeMapLayoutTraits Enumeration
Table of Predefined Shapes
Online Resources
Technical Support
MindFusion Products
Acknowledgements

Introduction

MvcDiagram is a general-purpose software component that can be used to display various types of diagrams and charts in web applications. It can be easily integrated into any project targeting the Microsoft ASP.NET MVC platform. By providing your software with ready to use capabilities for visualizing and editing relational data, MvcDiagram saves you from hundreds of hours of coding and debugging efforts.

MvcDiagram can give graphical representation to:

- Workflow processes
- Object hierarchies and relationships
- Graphs and networks
- Entity-relationship / database diagrams
- IVR systems
- Industrial automation processes
- Genealogy trees
- Algorithms and flowcharts
- Organizational charts
- XML documents
- Classification hierarchies
- Data mappings

MvcDiagram provides intuitive user-interaction model for creating or editing diagrams. In addition, its programmatic interface places at your disposal many useful functions, ranging from printing and XML serialization to path-finding and automatic layout algorithms.

What's New in this Release

The list below describes recent changes and additions to MvcDiagram:

New in version 4.6

CSS Styling

The styling system now supports defining item appearance through external CSS files. You can style different item types using custom element selectors such as **mfd-shapenode**, **mfd-diagramlink**, or **mfd-tablenode**. The system maps CSS Custom Properties to the internal Style properties. For example, use **--brush** for backgrounds, **--stroke** for outlines, and **--font-size** for text. The new cssClass property on DiagramItem allows for more granular styling using standard class selectors in your CSS. To ensure high performance, computed styles are cached globally, and the system automatically resolves style inheritance across class hierarchies (e.g., a ShapeNode will resolve styles from **mfd-shapenode**, **mfd-diagramnode**, and **mfd-diagramitem** selectors in that order).

CSS Copy Code

```
/* Style all shape nodes */
mfd-shapenode {
    --brush: #f9f9f9;
    --stroke: #333;
    --stroke-thickness: 1px;
    --font-name: 'Verdana';
}

/* Specific styling via the new cssClass property */
mfd-shapenode.urgent {
    --brush: #ffcccc;
    --stroke: #ff0000;
    --stroke-thickness: 2px;
}

/* Style table captions using custom properties */
mfd-tablenode {
    --caption-brush: #e0e0e0;
    --text-color: #000066;
}
```

Compact Multi-graph Layout

A new MinimalArea option has been added to the **MultipleGraphsPlacement** enumeration, which uses a bin-packing algorithm to arrange connected components in a compact, square-like area. This ensures a more efficient use of space compared to horizontal or vertical arrangements when laying out multiple independent graphs.

Polygonal Text Layout

The ShapeNode class now supports a polygonalTextLayout property, enabling text to wrap and align precisely within the geometric boundaries of non-rectangular shapes like triangles, diamonds, and custom polygons. This feature supports both plain and styled text across all alignment modes.

Version History

The list below describes past changes and additions to MvcDiagram:

New in version 4.5.1

Server-side Tag serialization

- Complex Tag and Id objects (class or struct instances) are now saved using **System.Text.Json.JsonSerializer** when saving a diagram to JSON and are automatically preserved during round-trips without having to handle serialization events. JsonSerializeTag and JsonDeserializeTag still allow you to implement custom serialization.
- Due to the removal of BinaryFormatter in .NET 9+, complex **Tag** and **Id** objects (class or struct instances) are now saved using **System.Xml.Serialization.XmlSerializer** when saving a diagram to XML. SerializeTag and DeserializeTag events still allow you to implement custom serialization.

Miscellaneous

- generateShortestRoute and findShortestRoute methods added to PathFinder.
- Table row anchorPattern fixes.
- Coordinates in vertical Ruler scale are now rotated in opposite direction to read from bottom to top, which is the accepted standard for rotated labels and consistent with MindFusion components for other platforms.

API changes

- **XmlSerializer** used to save Tag and Id objects could be a breaking change if the tag types are not serializable to XML out of the box. You can continue to serialize tags in binary format by setting the UseXmlSerializer property of XmlOptions to **false** (and adding the *BinaryFormatter* NuGet package to .NET 9+ projects).

New in version 4.5

Paged containers

PagedContainerNode is a ContainerNode subclass that organizes its child nodes into a collection of pages, represented by ContainerPage objects. This allows for the creation of more organized diagrams where nodes can be grouped into logical views within a single container. Only the nodes belonging to the currently active page, specified by currentPage property, are rendered and participate in user interactions.

Navigation between pages is handled by scroll arrows in the caption bar, allowing users to cycle through the pages sequentially. For faster access, users can also double-click the container's caption to open an in-place combo box listing available page titles, enabling direct navigation to any page. The node's text value is automatically set to the title of newly activated **ContainerPage**.

Client-side link routers

PatternRouter and CompositeRouter are now available in JavaScript front-end (previously only on server side / .NET). Diagram's default router is now set to a **CompositeRouter** instance containing a **PatternRouter** and Router sequence. This applies the several standard pattern paths when possible, and otherwise routes links using a simple cost-minimizing router.

Miscellaneous

- Improved BPMN shapes.
- Spatial indexing fixes.
- Visio2013Exporter improvements.

API changes

Default Diagram.linkRouter changed from Router to CompositeRouter instance.

New in version 4.4

ItemListView component

ItemListView supersedes NodeListView and lets you show both nodes and links in the list. Items can be added by calling AddItem method, and removed by calling RemoveItem. DefaultNodeSize property from **NodeListView** is called NewInstanceSize in the new class. The AutoConnectLinks and AutoConnectDistance properties control whether links created via drag-and-drop should connect to nearby nodes in the target diagram. **ItemListView** also provides more properties for customizing the list layout, such as Columns and LabelPosition. For compatibility with legacy code, **NodeListView** won't be removed from the library, bit it won't be getting any new improvements.

Palette component

The Palette control represents a tool palette that displays DiagramItem objects grouped into categories. It implements **Accordion** user interface that displays ItemListView components as its child panels, and category icon, title and collapse/expand button in pane headers. Each child **ItemListView** displays the items added to its corresponding category. Call the AddCategory method to define a category and create its respective accordion pane. Call AddItem to add an item to specified category. Palette provides same layout and appearance properties as the **ItemListView** class.

ItemLabel image

New Image and ImageUrl properties added to ItemLabel let you display icons along a link's geometry, or add more images to a ShapeNode. If a label contains both Text and **Image**, their relative position is specified by ImageAlign property (by default the image is placed on left side of text), and distance by ContentPadding. The image is rendered using bitmap's intrinsic size, unless you override it by setting ImageSize property. You can treat label images as clickable icons by handling linkClicked or nodeClicked events and checking label argument passed to their handlers.

Miscellaneous

- Locked property added to ItemLabel. It lets you prevent users from editing label's text, and moving it when Behavior is set to MoveLabels.
- label argument now provided with item click and double click events.
- Improved repaint speed of Overview component.

New in version 4.3.1

Node constraints

The constraints property lets you apply size and direction constraints to a node. It returns a NodeConstraints object with following properties:

- keepRatio: a bool value indicating whether the initial width/height ratio of a node should be preserved.
- maxHeight: the maximum node height allowed.
- maxWidth: the maximum node width allowed.
- minHeight: the minimum node height allowed.
- minWidth: the minimum node width allowed.
- moveDirection: a member of the DirectionConstraint enumeration. Set it to Horizontal or Vertical to let users move a node only in specified direction.

These constraints are considered when a user modifies the node interactively. They are ignored if node placement is changed programmatically by using API methods and properties.

Tag Helpers

The following tag helpers can now be imported from MindFusion.Diagramming.Mvc namespace:

- <mindfusion-diagramview> creates a DiagramView instance.
- <mindfusion-ruler> creates a Ruler instance.
- <mindfusion-zoomcontrol> creates a ZoomControl instance.
- <mindfusion-overview> creates an Overview instance.
- <mindfusion-nodelistview> creates a NodeListView instance.

Razor Copy Code

```
@addTagHelper *, MindFusion.Diagramming.Mvc
....
<mindfusion-diagramview
    component=(DiagramView)ViewBag.DiagramView
    style="width:100%; height:100%;">
</mindfusion-diagramview>
```

New in version 4.3

Multi-platform support

Multi-platform **System.Drawing** support has been deprecated in .NET 8. To allow hosting on non-Windows servers, we have moved server-side diagram code to **Microsoft.Maui.Graphics** API.

.NET 8 support

Added assemblies and sample projects for .NET 8. Distribution no longer includes assemblies for older .NET versions. If bug fixes are required, we will be patching version 4.2 branch for clients that host on old .NET versions, but no new features will be added to it.

Shape design enhancements

- The anchorPattern property of Shape lets you associate anchor points with shape definitions. That pattern is shared between all nodes of same shape, but can be overridden by setting node's own anchorPattern.
- Individual segments of a shape's geometry can be styled independently by setting stroke and fill attributes of ShapeElement and ShapeDecoration objects.
- We have replaced the Windows -only shape designer tool with an online shape editor at https://mindfusion.dev/tools/shape-designer.html
- The shape editor is implemented as a reusable ShapeDesigner control, which can be integrated into your own applications.

New client-side event system

Event emitter objects are now exposed as properties of Diagram and DiagramView, and you can register handlers by calling their addEventListener method:

JavaScript Copy Code

```
diagram.nodeCreated.addEventListener(
    (sender, args) =>
    {
        if (sender.defaultShape.id == "Rectangle")
            args.node.anchorPattern = pattern1;
        else
            args.node.anchorPattern = pattern2;
    });
```

Legacy event handling syntax will remain supported for compatibility with old code.

Collapse and expand table rows

Table rows can be assigned to distinct sections of the table. Each section can be collapsed or expanded, hiding or showing all rows in the section except the header one. To define a section, set the header property of a row. A section consists of all rows after a header row and spans to the next header. Each header row displays a [±] button that allows expanding or collapsing the section interactively. Clicking that button raises the tableSectionCollapsed and tableSectionExpanded events. Sections can be collapsed or expanded programmatically by setting the expanded property of their header rows.

New mouse events

- **nodePointed** event has been replaced by a set of more specific events: nodePointerEnter and nodePointerLeave are raised when the mouse pointer respectively enters or leaves the boundaries of a node. nodeHovered fires if the pointer rests over a node for hoverDelay milliseconds.
- **linkPointed** event has been replaced by a set of more specific events: linkPointerEnter and linkPointerLeave are raised when the mouse pointer respectively enters or leaves the boundaries of a link. linkHovered fires if the pointer rests over a link for **hoverDelay** milliseconds.

Miscellaneous

- dividerStroke properties added to ContainerNode, TableNode and TreeViewNode classes. It lets you customize the appearance of the divider line drawn between a node's caption bar and content area.
- Set the autoDeleteLinks property to **false** to prevent deleting links along with a node they connect, and disconnect the links instead.
- Enable the autoDeleteChildren property to automatically delete child nodes when a container is deleted.
- Removed an **eval** call used for creating custom components from CompositeNode Json templates, allowing for stricter Content Security Policy rules. Now custom components must be registered by calling registerComponent method to allow instantiating them from a template.
- linkRouted event raised when links are modified by automatic routing.

API changes

- To allow for stricter Content Security Policy rules, the library no longer loads CSS scripts automatically. If you use components from the Diagramming.Controls package, you must now explicitly link the common-ui.css file from distribution's *themes* folder.
- **linkPointed** and **nodePointed** events replaced by linkHovered and nodeHovered.

.NET API changes

- All properties and method parameters of **RectangleF**, **PointF**, **SizeF** and **Color** types from **System.Drawing** namespace have been changed to respective **Rect**, **Point**, **Size** and **Color** from **Microsoft.Maui.Graphics** namespace.
- Due to **BinaryFormatter** deprecation and runtime exceptions in non-desktop applications, binary **SaveToFile** and **LoadFromFile** methods have been removed. Remaining serialization methods allow storing diagrams in JSON and XML formats.
- **SwfExporter** (Adobe Flash format) and **WmfExporter** (Windows Metafile) classes have been removed.

New in version 4.2

Multiple diagram pages

The DiagramDocument class represents a collection of diagram pages or sheets, represented by DiagramPage objects. New pages can be added to the document and existing pages can be removed and reordered by modifying the Pages collection.
The SaveToJsonFile, SaveToXml and SaveToString methods of **DiagramDocument** save all pages in a single file, and respectively the LoadFromJsonFile, LoadFromXml and LoadFromString methods load all pages from the file.

DiagramPage is derived from Diagram and only adds a few properties, so any code that processes **Diagram** objects will work with **DiagramPage** objects too. A diagram document forwards each event raised by its pages, giving you a chance to use the same handler for events raised by different pages. If a page should implement distinct user interaction or validation logic, it is also possible to attach handlers to the **Diagram** events inherited by **DiagramPage**.

A single **DiagramPage** could be assigned to the Diagram property of DiagramView, and that lets you create your own user interface for presenting a list of pages to the user and selecting a current page. The package also includes a TabbedDiagramView that presents all pages of a document in a tabbed user interface where each tab corresponds to a page. The page displayed in the active tab is exposed by the selectedPage property of TabbedDiagramView. The active page can be changed interactively by activating its associated tab. New pages can be added and removed to / from the document through buttons in the tab strip and the pages can be rearranged by dragging their associated tabs with the mouse.

Tabbed diagram view

The TabbedDiagramView control is a view that displays DiagramDocument objects. Individual diagram pages can be activated through the tabs located at the top or at the bottom of the control. The appearance and behavior of the tab strip can be customized using various properties. The location and visibility of the strip is specified through the TabAlignment property. Its height can be changed via the TabsSize property. ShowAddButton and ShowNavigationButtons determine the visibility of the buttons for

interactive tab creation and tab navigation respectively. ShowTabCloseButtons specifies whether the tabs can be closed interactively, by clicking on the small 'x' button displayed in each tab. AllowReorder enables interactive tab rearrangement by mouse dragging.

Async serialization

Files can now be saved and loaded asynchronously. Async methods create a copy of the diagram to process it in a worker thread, and custom item classes must implement Clone method or copy constructor in order to serialize them as expected.

- async SaveToFileAsync and LoadFromFileAsync methods of Diagram and DiagramDocument implement serialization in binary format.
- async SaveToXmlAsync and LoadFromXmlAsync methods of **Diagram** and **DiagramDocument** implement serialization in XML format.
- async SaveToJsonFileAsync, SaveToJsonAsync, LoadFromJsonFileAsync, LoadFromJsonAsync methods of **Diagram** and **DiagramDocument** implement serialization in JSON format.

Reversed tree layouts

(client side / JavaScript)

The reversedLinks property of TreeLayout, RadialTreeLayout and BorderedTreeLayout lets you arrange trees where links point from child to parent nodes, instead of the default parent to child direction.

Miscellaneous

- column, row and node property getters added to CellEventArgs.
- fixed RaiseEvents mode of expandButtonAction; it no longer collapses or expands clicked node's subtree.
- ContainerNode now sorts its child nodes by zIndex instead of insertion order.

New in version 4.1.1

Client-side radial layout

The RadialTreeLayout class arranges tree levels in concentric circles around the root. The levelDistance property specifies the distance between adjacent circles. The direction property specifies placement of first child node. Set stretchFactor to a value larger than 1 in order to stretch level circles into ellipses, providing more space for the layout when using wider nodes.

Miscellaneous

- strokeDashArray and strokeDashOffset properties let you set custom dash patterns when strokeDashStyle is set to Custom.
- Styled text now supports <size=N> element letting you set font size.
- enableStyledText property added to ItemLabel lets you format labels using HTML-like format flags.
- enableWrap and maxWidth properties of **ItemLabel** let you wrap label's text.
- The ArrowHeads class exposes arrowhead shapes as static methods returning respective predefined shape.
- Fixed linkTextStyle exception.
- Fix for NodeListView not working correctly when hosted nodes contain NodeLabel objects.
- New PatternRouter constructor accepts string definitions of link routes.
- EnableParallelism and MaxDegreeOfParallelism layout properties now available in CoreCLR assemblies (.NET Core 3 and 5+ versions).
- New Export overloads of exporter classes allow writing to **System.IO.Stream**.

New in version 4.1

Multiple labels per node

The NodeLabel class allows multiple captions to be displayed for a single DiagramNode of any type. Node labels provide a set of properties allowing full customization of their display and positioning. Label position is defined by specifying a pin point and offset from it, set through SetCornerPosition, SetEdgePosition, SetCenterPosition methods. In addition, the HorizontalAlign and VerticalAlign properties of **NodeLabel** specify on which side of pin point to draw the label's caption.

For example, following code adds extra labels to top-left and bottom-right corners of a ShapeNode:

C# Copy Code

```
var node = diagram.Factory.CreateShapeNode(10, 50, 40, 30);
node.Text = "text"; // centered main text

var lb1 = node.AddLabel("label 1");
lb1.SetCornerPosition(0, 0, 0);
lb1.HorizontalAlign = StringAlignment.Near;
lb1.VerticalAlign = StringAlignment.Near;
lb1.Brush = new SolidBrush(Color.Red);
lb1.TextBrush = new SolidBrush(Color.White);

var lb2 = node.AddLabel("label 2");
lb2.SetCornerPosition(2, 0, 0);
lb2.HorizontalAlign = StringAlignment.Far;
lb2.VerticalAlign = StringAlignment.Far;
lb2.Brush = new SolidBrush(Color.Yellow);
lb2.TextBrush = new SolidBrush(Color.Red);
```

Multi-touch improvements

- The MultiTouchDraw property of DiagramView lets you prevent drawing multiple items simultaneously, while keeping other multitouch gestures enabled.
- The HandleTouchHitDistance property of **DiagramView** makes it easier to grab adjustment handles on mobile devices, without increasing the AdjustmentHandlesSize value.

Miscellaneous

- The MoveLabels behavior lets user move link and node labels interactively.
- HorizontalOffset and VerticalOffset now work for link labels positioned using SetLinkLengthPosition method.
- ExcelExporter exports pen width of links.
- **ExcelExporter** sets more precise positions of link end points.
- LinkEventArgs now includes origin and destination properties that report respective candidate nodes to LinkCreating and LinkModifying client-side event handlers.
- SvgNode.ContentAlign property specifies alignment of SVG drawing inside the node.
- DiagramLink.lineAlignment property sets the vertical position of link's text.
- Horizontal scroll gestures should now work on Macbook trackpad when using virtual scroll mode.

New in version 4

ES6 classes and properties

(client side / JavaScript)

Client side scripts have been refactored to use ES6 classes, properties and native collections. **ArrayList**, **Dictionary** and **Set** classes from MindFusion.Collections namespace have been removed, and replaced by respective JavaScript native **Array**, **Map** and **Set**. Get/set functions have been replaced by getter/setter properties, which should help using the diagram API with binding expressions of various JS frameworks. You could enable propFunctions in CompatConfig in order to continue using legacy get/set functions during migration.

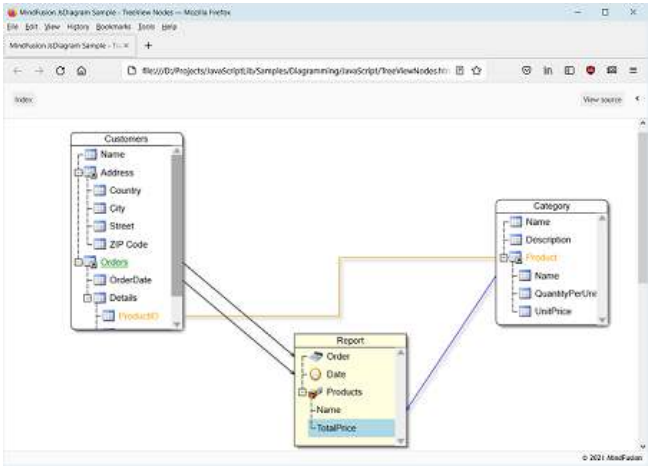
Diagram view

(client side / JavaScript)

The Diagram JavaScript class has been refactored into separate **Diagram** and DiagramView classes, matching respective server-side .NET classes that the library already provided in older versions. What remains of **Diagram** is now just a model class that defines diagram structure and implements serialization. The new **DiagramView** class deals with rendering and user interaction. You can show the same diagram in several views, each one rendering at different scroll positions and zoom levels. Note that you can no longer call *Diagram.find(id)* on client-side, but instead call *DiagramView.find(id).diagram* to access the **Diagram** object.

TreeView nodes

The TreeViewNode class represents nodes that can display hierarchical data. The root items displayed in the node can be accessed through the rootItems property. Items can be added and removed individually by using the addItem and removeItem methods, or in bulk by calling the fromObject method, which loads the tree view items from an array of objects.



Print pagination

(client side / JavaScript)

The printPreview and print methods of DiagramView let you export the diagram as a list of smaller images in HTML page. Supported options include **printArea** (defaults to diagram's content bounds) and pageSize (defaults to DiagramView's viewport). Note that print methods use HTMLCanvasElement.toDataURL() internally, so any limitations it has will apply (such as canvas size and CORS).

.NET 6 support

Added assemblies and sample projects for .NET 6. The distribution still includes assemblies and samples for .NET 4.5+ and .NET Core 3.1.

Multi-touch support

- If MultiTouchZoom property is enabled (default), the view can be zoomed or panned using two-touch pinch / flick gestures.
- If MultiTouchModify property is enabled (default), diagram nodes can be moved, scaled and rotated using two-touch pinch / flick gestures.
- If **MultiTouchZoom** property is disabled, each touch draws diagram items corresponding to current behavior.
- If **MultiTouchModify** property is disabled, each touch started from a node draws a diagram link.

- Latter modes can be used for collaborative whiteboarding / classroom scenarios.

Mouse input improvements

- MouseWheelAction property of DiagramView lets you choose between Scroll (default) and **Zoom** as the default behavior of the control in response of a wheel event.
- The MoveNodes behavior allows grabbing nodes to drag them without using adjustment handles.
- The control now captures mouse events and completes drag / drawing operations even if mouse is released outside the diagram view.

New events

- NodeSelectingScript, LinkSelectingScript, SelectionChangedScript events allow handling of selection interactions.
- NodeTextEditingScript, LinkTextEditingScript and CellTextEditingScript validation events let you prevent users from editing a specific item
- DataLoadedScript is raised when the diagram data is loaded from JSON or XML.
- MouseWheelScript is raised when the user rotates the mouse wheel while the cursor is positioned over the diagram surface.
- TreeItemTextEditingScript, TreeItemTextEditedScript allow handling of inplace-edit operations in TreeViewNode.

Miscellaneous

- Client side createDiagramLink method now accepts two, three or four arguments, letting you specify row index for origin or destination TableNode, or TreeViewItem for origin or destination TreeViewNode.
- OrthogonalLayout can now be applied on client-side.
- Clipboard methods now use modern **Navigator.clipboard** API when their *systemClipboard* argument is enabled.
- Rotation of FreeFormNode instances.
- ImageAlign supports new **FitLeft**, **FitTop**, **FitRight** and **FitBottom** alignment styles, which resize image to fit node's boundaries and align it to respective border.
- Fix for resizeToFitText not calculating correct size when cells have text padding applied.
- ArrangeAsync method is now available in CoreCLR assemblies (.NET Core 3 and 6 versions).
- The EnumAllPaths method of PathFinder yields each path immediately when found.

JavaScript API changes

- getProperty/setProperty function pairs from client-side API have been replaced by ES6 getters/setters. Set CompatConfig.propFunctions to **true** in order to continue using legacy get/set functions in your code.
- **Diagram.find** method has been removed. Call DiagramView.find instead, and get the view's diagram property.
- Client-side scroll and zoom properties moved from Diagram to DiagramView: autoScroll, scrollZoneSize, autoScrollAmount, scrollX, scrollY, viewport, virtualScroll, zoomFactor.
- Client-side input related properties moved from **Diagram** to **DiagramView**: allowInplaceEdit, delKeyAction, behavior, tooltipDelay, modificationStart, modifierKeyActions, leftButtonActions, middleButtonActions, rightButtonActions.
- Client-side magnifier properties moved from **Diagram** to **DiagramView**: magnifierEnabled, magnifierFactor, magnifierWidth, magnifierHeight, magnifierFrameThickness, magnifierShading, magnifierShape,magnifierFrameColor, magnifierSecondaryFrameColor.
- Client-side methods moved from **Diagram** class to **DiagramView** class: create, **find**, record, stopRecording, replay, clearTooltip, beginEdit, copyToClipboard, pasteFromClipboard, cutToClipboard, scrollTo, zoomToRect, zoomToFit, setZoomFactorPivot.
- **MindFusion.AbstractionLayer** class has been removed.
- EventArgs and CancelEventArgs classes have been moved to the MindFusion.Controls namespace.
- **ArrayList**, **Dictionary**, **ObservableCollection**, **Set** classes have been removed from MindFusion.Collections namespace.
- Due to **Navigator.clipboard** API being asynchronous, pasted items are not available immediately after calling **pasteFromClipboard** with *systemClipboard* enabled. You will be able to access them only after nodePasted and linkPasted events are raised.

.NET API changes

- Server-side ContainerNode API is now consistent with client-side one. Containment is no longer implemented via a Group object and SubordinateGroup / MasterGroup properties. Instead, child nodes are now stored in the dedicated Children collection, and a node's container is returned by the Container property.
- **Suspend** and **Resume** methods removed from ILinkRouter. Call Diagram's SuspendLinkRouter and ResumeLinkRouter methods instead.
- Default LinkRouter changed from QuickRouter to CompositeRouter instance.
- bool **SmartPolylineEnds** property of RoutingOptions replaced by PolylineRouting enum property, which also provides an option to prevent inclined link segments for Polyline routes.
- Removed **ClientSideMode**, **MsAjaxLocation** and **JQueryLocation** properties.

New in version 3.3.1

Adjustment handles styling

Appearance of adjustment handles can be customized via ActiveItemHandlesStyle, SelectedItemHandlesStyle and DisabledHandlesStyle properties. The HandlesVisualStyle objects returned by them provide sub-properties corresponding to graphic attributes of the different handle types. Adjustment handles can now be painted not only in solid color but with arbitrary brushes such as LinearGradientBrush.

Miscellaneous

- TopologicalLayout can now be applied on client-side.
- The diagram canvas now automatically expands to fill the area freed up when a scrollbar auto-hides.
- Fix for scroll position serialization in virtual scroll mode.
- ActiveItem reference is now serialized in JSON format.
- Fixed Overview flickering when the tracker rectangle is being dragged near the control's borders.
- Client-side XML serialization fixes.
- Fix for shadow style serialization in .NET Core and .NET 5.

New in version 3.3

.NET 5 support

MindFusion.Diagramming for ASP.NET MVC distribution now includes assemblies for .NET 5. The programming interface is the same as the one of .NET Core Assemblies.

Control nodes

ControlNode objects display custom HTML content, specified via their Template property. A node's DOM structure is created inside the Content div element, which is then rendered on top of the diagram canvas. Note that this prevents **ControlNode**s drawing in diagram' usual Z order, and they will always appear on top of other diagram items.

By default the diagram intercepts mouse events of all hosted html elements to enable moving nodes or drawing links. You can specify that interactive elements, such as buttons or text boxes, should receive input instead by setting their **data-interactive** attribute to **true**.

You can register event handlers for elements in the template by assigning function names to **data-event** attributes in the form data-event-'eventName'='handlerName'. Alternatively, you can attach event listeners using DOM API once nodeDomCreated is raised to signify that **content** is available. E.g. access child elements by calling args.getNode().getContent().querySelector(...) and attach handlers via elements' **addEventListener** method.

ControlNodes create DOM elements from their template only in the main diagram, and render images instead when shown in Overview or NodeListView. Note that in this case image elements in the template will render only if specified in base64 format. The Utils.toDataUrl method helps you convert an image URL to base64 data URL.

Video nodes

VideoNode objects display video streams, along with UI elements to play, pause, seek and change volume. Call node's setVideoLocation method to specify its video URL. Auto-playing is not supported at this time, and the video can be started only by users clicking the play button. **VideoNode** is built around Video component, which can also be used in CompositeNode templates.

Button components

Composite nodes can now contain buttons. Button components respond to user clicks by calling the JavaScript function whose name is assigned to their **clickHandler** attribute. A button can show a text label assigned to the **text** attribute, or an image whose URL is assigned to **imageLocation**. The following example adds a button to CompositeNode template:

JavaScript Copy Code

```
{
    component: "Button",
    brush: "#ccc",
    text: "Delete",
    width: 30,
    cornerRadius: 3.5,
    clickHandler: "onDeleteClick"
}
```

Virtual scroll improvements

- The control now handles mouse wheel events to scroll the diagram when virtual scrolling is enabled.
- DOM elements that implement virtual scroll mode now use CSS grid layout. You might need to adjust page layout to apply size constraints on the diagram div or its parent. For compatibility with older browsers, you can set CompatConfig.gridLayout = **false** to fall back to the absolute positioning from previous versions.
- VirtualScroll mode is now enabled by default.

Miscellaneous

- Mouse pointer position is now provided as argument to nodeCreated and linkCreated events.
- Visibility property added to CompositeNode components lets you keep invisible components in the template.
- General performance improvements.
- Collapse / expand icon is now drawn correctly for **CompositeNode** when Expandable is enabled.
- Json deserialization no longer creates undo records.
- Content of **CompositeNode**s is now clipped by default if Shape in the template is marked with **isOutline** attribute. To disable that, set ClipToOutline to **false**.
- Set ExpandButtonAction to RaiseEvents and +/- buttons of Expandable nodes will only raise an expandButtonClicked event instead of expand or collapse the tree under that node. You can use this to implement custom logic for the +/- button.

New in version 3.2.2

.NET Core support

MindFusion.Diagramming for ASP.NET MVC distribution now includes assemblies for .NET Core 2.1 and newer versions. The programming interface is mostly the same, except for a few differences listed in .NET Core Assemblies help topic. Tutorial help topics have been updated for .NET Core as well.

Shape components

The Shape class used to specify ShapeNode geometry can also be used as a component in CompositeNode. When its **isOutline** attribute is set, the shape will control **CompositeNode**'s geometry too, defining hit test and clip area, and link alignment points along node's border. If **isOutline** is disabled, the shape will serve mostly as a decorative element. The code below shows a sample fragment from node's template specifying shape component's properties:

Json template

Copy Code

```
{
    component: "Shape",
    id: "Cloud",
    autoProperty: true,
    name: "OutlineShape",
    pen: "gray",
    brush: "Transparent",
    isOutline: true
},
```

Miscellaneous

- TableNode.resizeToFitText improvements and fixes.
- containerChildAdding event handlers can stop propagation of the event up in the container hierarchy by calling setHandled.
- LayeredLayout now uses stable sorting and should always create same results when applied to same graph.
- For consistency with client side API, the .NET Diagram class now provides Arrange methods that take a layout object as argument.
- Arrange method of .NET ContainerNode arranges the child items inside a container using the specified layout class.

API changes

- The AutoJson attribute has been moved from MindFusion.Diagramming.Mvc namespace to MindFusion.Diagramming namespace.
- LoadFromJson now clears the diagram by default. A new LoadFromJson overload lets you specify whether loaded items are merged into current diagram or the old diagram is cleared first.

New in version 3.2.1

Improved performance

Diagram rendering and user interactions are now faster. The control redraws the smallest possible area after diagram changes, and compresses refresh operations into a single batch when possible. Custom node classes that need to draw outside of their Bounds should override the getRepaintBounds method to specify the repaint area.

Custom type registration

Custom item types registry is now stored in the session and is handled by the ItemTypeResolver class. It is no longer required to call the RegisterItemType method
for every DiagramView and NodeListView instance, but instead call **RegisterItemType** method only once on the first **DiagramView** control and register the current session by calling its RegisterSession method. Then call the RegisterSession method for any subsequent controls that will use the custom item types. You can access the **ItemTypeResolver** registered for the current session through the static GetResolver method and unregister it by using the Unregister static method.

Miscellaneous

- The Border component draws frame lines around and background behind other components in CompositeNode templates.
- NodeListView now displays nodes' ToolTip texts when the mouse hovers over an item in the list view.
- Fix for FlowchartLayout creating node overlaps in some situations.

New in version 3.2

Flowchart graph layout

(client side / JavaScript)
FlowchartLayout recognizes program code-like patterns in the graph, such as loops, sequences and if/switch branchings, and arranges them recursively. **FlowchartLayout** could be used to arrange other types of graphs as well, though with some restrictions. For example it treats all back links as loops in the code, and expects that they are nested - loop links starting closer to the stop node should end closer to the start node. Another similar restriction is that there shouldn't be any cross-links that connect different branches of a decision sub-graph.

Path finding

(client side / JavaScript)
The PathFinder class provides methods that help you find paths and cycles in a graph:

- findShortestPath finds the shortest path between two DiagramNode objects.
- findLongestPath finds the longest path between the specified **DiagramNode** objects.
- findCycle detects whether the specified **DiagramNode** participates in a cycle.
- findAllPaths finds all paths that exist between two **DiagramNode** objects.
- findAllCycles finds all cycles in the underlying diagram.

Path objects returned by these methods contain nodes, links and items arrays containing sequences of elements in the path. The new PathFinder sample page included in distribution demonstrates path finding and animations over the found paths.

Embedded hyperlinks

Nodes and Text components with style text enabled can now contain <a> tags to create hyperlinks. When a link is clicked, the control raises hyperlinkClicked event to let you implement navigation:

JavaScript

Copy Code

```
// node is a ShapeNode instance
node.setText("test <a='https://mindfusion.dev'>link</a> test");
node.setEnableStyledText(true);

// attach an event listener to the hyperlinkClicked event
diagram.addEventListener(Events.hyperlinkClicked, onHyperlinkClicked);

function onHyperlinkClicked(sender, args)
{
    window.open(args.getHyperlink());
}
```

Container improvements

- resizeToFitText method and EnableStyledText property added to the ContainerNode class.
- containerChildRemoved event now also raised when dragging multiple selected child nodes out of a container.
- Improved handling of folded containers by automatic layout classes and link routing methods.

Visio2013Exporter improvements

- Visio2013Exporter now does full rendering for table, container and composite nodes.
- **Visio2013Exporter** renders custom nodes that draw themselves through the IGraphics interface.
- Coordinates of exported link end points are now assigned Visio formulas and follow the borders of resized nodes.

Miscellaneous

- The Text property of nodes and links has been moved to the base DiagramItem class, allowing for easier text search and edit operations in the diagram. The Caption property of container and table nodes now only wraps the base **Text** property. Map and Visio nodes now render **Text** too.
- Typescript definitions now available for the Diagramming.Lanes namespace.
- Auto-arranged link labels now can optionally be placed over link segments and containers by setting diagram's AutoArrangeAvoidSegments and AutoArrangeAvoidContainers properties to **false**.
- The dispose method of Diagram class now removes a focus-proxy DIV element from the page.
- Improved text alignment for DiagramLink texts.
- The serializeTag event lets you save complex Tag and Id objects in XML format.
- Various arrowhead rendering fixes.

API Changes

FromJson methods now require id argument.

New in version 3.1

Composite nodes

The CompositeNode class implements nodes whose appearance can be defined via composition of components and layout containers. The content of a composite node can be created by building a tree of components programmatically, or by loading a JSON template. This initial release includes layout containers such as StackPanel and GridPanel. Objects from the MindFusion.Drawing namespace now double as **CompositeNode** components (Image, Text, Path, Rect). Future versions will add interactive components like buttons and text editors.

Animated layout

arrangeAnimated methods added to Diagram and ContainerNode classes display movement of items from their original locations to new locations assigned by the layout object. This can be used to create an explode effect by starting new layout from overlapping nodes on same position, or an insertion effect by adding a new node to an existing layout.

Export Excel files

The ExcelExporter class from *MindFusion.Diagramming.Export.Excel.dll* assembly exports diagrams to Excel Open XML format (XSLX) files. Diagrams are exported as Excel AutoShapes drawings. The Export overload that takes a DiagramDocument parameter creates a sheet for each DiagramPage in the document. The Export(Diagram) overload creates a single Excel sheet.

Overview improvements

- ScaleMode enumeration and properties added to Overview control implement several scaling modes. The FitAll element corresponds to the original behavior where the Overview always fits diagram's contents. **FixedScale** mode applies the scale specified via overview's ScaleFactor property. **CombinedScales** mode multiples overview's **ScaleFactor** by diagram's current ZoomFactor.
- If AllowZoom is enabled, users can zoom the diagram by resizing the overview's viewport tracking rectangle.
- MinVisibleFontSize specifies a threshold value that hides text if scaled font sizes become smaller.

Visio2013Exporter improvements

- Visio2013Exporter processes links' arrowheads.
- **Visio2013Exporter** preserves the diagram Z order.
- **Visio2013Exporter** exports links' Text and Labels.
- **Visio2013Exporter** exports Image of ShapeNode objects.

LayeredLayout improvements

- The KeepLinkShapes property of LayeredLayout lets you preserve initial Shape value of links when LinkType is set to Cascading. The orthogonal path applied by latter property is approximated using respective link shape.
- Fix for wrong link positions when the diagram contains folded containers.

- Fix for unnecessarily large distance between subsequent layers after a long node appears in a layer.
- LinkPadding property added to **LayeredLayout** specifies how much space to leave between links when **LinkType** is set to **Cascading**.

PathFinder improvements

- Speed of FindAllCycles and EnumAllCycles methods improved.
- **FindAllCycles** and **EnumAllCycles** no longer report mirrored versions of cycles in undirected mode.

Miscellaneous

- ImagePadding property of ShapeNode and Cell lets you set padding space between elements' borders and their Image.
- Export overloads that write to **System.IO.Stream** objects added to PdfExporter, VisioExporter and Visio2013Exporter.
- Non-blocking ArrangeAsync methods run layout calculations in a background task. This is only available in .NET 4.5 build of the diagram assemblies.

New in version 3

User-interaction controllers

All kinds of user interaction are now carried out by controller objects. The **startDraw** method of BehaviorBase-derived classes that returned an opaque **InteractionState** instance has been replaced by createController method returning a controller of class specific to the modified item and interaction types. Built-in controllers include CreateNodeController, CreateLinkController, ModifyNodeController, ModifyLinkController, PanController. You can also create custom controllers by deriving and overriding the methods of SinglePointerController:

JavaScript Copy Code

```
SinglePointerController.prototype =
{
    start: function (position) {},
    move: function (position) {},
    validate: function (position) {},
    commit: function (position) {},
    cancel: function (position) {},
    drawInteraction: function (context) {}
}
```

Record and replay

The Diagram class provides record, stopRecording and replay methods that can be used to save and replay all user actions with keyboard and mouse. The replay method takes *recording*, *timer* and *diagramJson* parameters. The *recording* argument is an array of records returned by stopRecording. If the *timer* argument is set to **true**, events are replayed from a timer preserving their original timing, otherwise they are replayed immediately. Timer mode can be used to show tutorials played directly in the diagram. Immediate mode could be used to automate user interaction testing where an assertion is checked after replaying series of events. Recorded events could also be replayed as macros if ran on a diagram with items placed on same initial coordinates.

Improved testability

User interaction can now be tested automatically in several ways outlined below.

- Create a MouseInputDispatcher instance and call its methods

  JavaScript Copy Code

  ```
  onMouseDown: function (position, button)
  onMouseMove: function (position)
  onMouseUp: function (position, button)
  ```

  where position is a Point instance specifying logical coordinates in the diagram (in current measure unit, not scaled by zoom factor). The results of called mouse methods will depend on diagram's current Behavior, e.g. you could assert that a sequence of down, move and up events creates a ShapeNode when Behavior is set to DrawShapes.
- Create an instance of controller class and call its start, move and commit methods. This would not depend on the value of diagram's Behavior, but could be still affected by other behavioral properties such as AllowSelfLoops and AllowUnconnectedLinks.
- Check assertions after replaying a recording on a saved diagram.

Ruler control

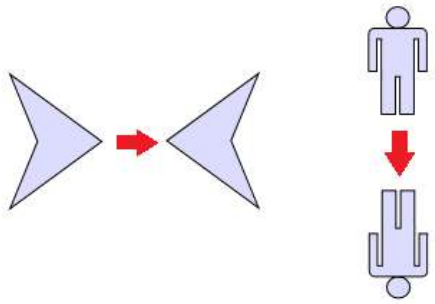
The Ruler control provides horizontal and vertical scales that help users measure and align diagram items.

3rd party dependencies now optional

The diagram can now be used without JQuery and Microsoft Ajax libraries. To enable that mode, set **ClientSideMode** to **StandAlone** for DiagramView. Stand-alone mode requires relatively modern browsers, the base feature-set against we test it is the one of IE11.

Flip shapes

The shape of a ShapeNode can be flipped horizontally or vertically. Enable the AllowFlip property and the shape flips when an edge is dragged over its opposite edge. You can get or set current flip state using FlipX and FlipY properties.



Revamped item dependency system

Dependencies between diagram items are now expressed by rule objects and applied in sequence by the DiagramMediator class. Built-in dependencies are implemented by ItemFollowsContainerRule, AttachedFollowsMasterRule, LinkFollowsNodesRule, NodeFollowsSelectionRule rules. You can create custom rule objects with following API and add them to DiagramMediator.AllRules array to make items follow other items during user interaction -

JavaScript — Copy Code

```
var MyRule =
{
    dependencies: function (master) {}, // return array of items that should follow master
    dependentUpon: function (dependency) {},  // return array of items that should be followed by
dependency
    recognize: function (dependency) {},  // return true or false if this rule should be applied
on an item
    updateFromMasters: function (dependency, originalStates) // set new position or size of
dependency
}
```

Miscellaneous

- CaptionBackBrush and FoldIconSize properties added to ContainerNode.
- Multiple rotation of nodes improved, now node angles are modified by same rotation delta calculated from mouse position instead of pointing all rotation handles in mouse direction.
- Xml serialization improved.
- LinkSegments property of Diagram specifies initial number of link segments.

New in version 2.8

Fluent API

Extension methods in MindFusion.Diagramming.Fluent and MindFusion.Diagramming.Layout.Fluent namespaces add support for fluent programming style:

C# — Copy Code

```
using MindFusion.Diagramming.Fluent;
using MindFusion.Diagramming.Layout.Fluent;
//...

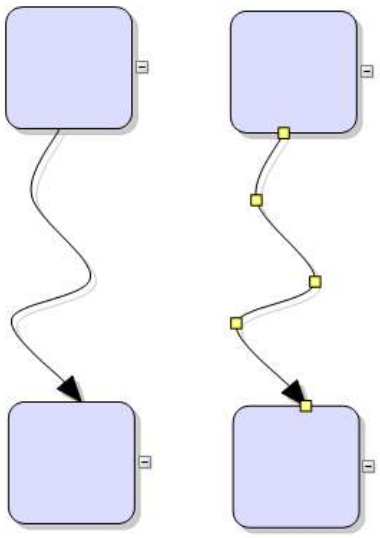
diagram.Factory
 .CreateShapeNode(10, 10, 20, 20)
  .Brush(Color.LightGray)
  .Font("Arial", 12)
  .EnableStyledText(true)
  .Text("Task <i>1</i>")
  .ToolTip("This is the task");

new TreeLayout()
 .LevelDistance(20)
 .NodeDistance(20)
 .LinkStyle(TreeLayoutLinkType.Cascading3)
 .Arrange(diagram);
```

DiagramLink improvements

- The component no longer keeps a separate segmentCount field, removing a common source of errors. The SegmentCount property now calculates its value from ControlPoints elements. The UpdateFromPoints(updateGroups, updateSegments) overload has been removed too.
- **SegmentCount** setter no longer refuses changing number of segments if auto-routing is enabled or the link is a self-loop.
- The HeadPen property lets you customize arrowhead strokes independently of line segments strokes.

- The new **Spline** element of LinkShape enumeration draws links as interpolating splines that pass through all of their control points:



Miscellaneous

- CaptionFormat properties of TableNode and ContainerNode let you set alignment for table and container captions.
- Enable the AllowResizeHeaders property of swimlanes Grid to let users resize grid rows and columns interactively. The control raises HeaderStartResizingScript validation event when the user tries to resize, and HeaderResizedScript event when the operation completes.
- NodeSize and DefaultNodeSize properties of NodeListView can be set to **SizeF.Empty**, in which case the control will use the sizes stored in nodes' Bounds.
- The control now supports cyclic attachment created using DiagramNode.attachTo method, where moving any of the attached nodes will move all other nodes in the cycle.
- Shapes.RoundRect corner arcs keep constant radius and no longer deform when the node size changes to non-rectangular.
- ItemAddedScript and ItemRemovedScript events raised when an item is added or removed, either from code or interactively.

New in version 2.7

Free-form nodes

A FreeFormNode collects all points from users' mouse or touch input and displays them as node's outline. To let users draw free-form nodes interactively, set Behavior to DrawFreeForms or **LinkFreeForms**. Use the Points property of **FreeFormNode** to get or set outline points programmatically. If the Closed property is set, the node is drawn as a closed shape and its interior filled, or otherwise the node is drawn as a poly-line. If the distance between first and last points drawn by user is shorter than AutoCloseDistance, the node's Closed property is automatically set to **true**. AutoCloseDistance default value is Number.MAX_VALUE, so free-form nodes are always closed.

Convert free-form drawings to ShapeNodes

Additional drawing modes, convenient for touch input, convert FreeFormNode objects drawn by user to ShapeNode objects with matching shapes. To enable them, set Behavior to DrawFreeShapes or **LinkFreeShapes**. The shapes against which the user's input is matched are set via diagram's FreeFormTargets property. By default it contains Rectangle, Decision and Ellipse shapes.

Miscellaneous

- TypeScript definitions for the Diagramming API are now provided in *jsdiag.d.ts* file.
- The MaxJsonLength and RecursionLimit static properties of DiagramView class let you configure the corresponding settings of internal **JavaScriptSerializer** instances.
- PdfExporter now creates page hyperlinks if HyperLink is set to a value in "page://{number}" format, e.g. set "page://1" to open the second PDF page when the item is clicked
- VirtualScroll mode fixed for parent div size specified in percent units.
- Fixed bug where the ExpandOnIncoming property specifying link direction for collapse/expand actions did not work correctly.
- Removed several global variables created inadvertently by the control.

New in version 2.6.1

- NodePastedScript and LinkPastedScript client-side events raised when items are pasted from the clipboard.
- **System3D** member added to CellFrameStyle enumeration. If it's assigned to the CellFrameStyle property of a TableNode, cell frames are rendered using 3D effect.
- Resize and move interaction cursors now remain visible until the user releases mouse button.
- Fixed bug where auto-handles mouse hint cursor would not match the item type.
- Fixed bug where multiple lines in link's Text were drawn overlapped.
- Fixed bug where BaseShape arrowhead would point in wrong direction for strictly vertical links.

New in version 2.6

Resize table columns and rows

Columns and rows of a TableNode can now be resized interactively if its AllowResizeColumns or AllowResizeRows properties are enabled. In order to resize, move the mouse pointer to the border line on column's right side or row's bottom side until it shows resize cursor and start dragging. The control raises tableColumnResizing and tableRowResizing events to let you validate new size or prevent resizing some elements. The tableColumnResized and tableRowResized events are raised after the operation completes.

Shape libraries

The ShapeLibrary class lets you use custom geometric shapes created using MindFusion ShapeDesigner tool. Call the ShapeLibrary.loadFromXml(url) method to load a library XML file. ShapeLibrary.getShapes() returns an array of the loaded Shape objects. The NodeListView.setShapeLibraryLocation(url) method creates a prototype ShapeNode object for each shape from the specified library. Location of a shape library can also be specified server-side through the DiagramView's ShapeLibraryLocation property.

Miscellaneous

- TextStroke and TextStrokeThickness properties of DiagramItem let you set color and thickness of text outlines;
- Items can now be deleted using Backspace key when running on Mac;
- Caption divider line in TableNode and ContainerNode is now rendered clipped when caption height is smaller than corner radius;
- Diagram.TooltipDelay property specifies the delay in milliseconds before showing tooltips;
- The NodeListView.Orientation property lets you set the view's orientation to Horizontal or Vertical (members of **MindFusion.Diagramming.Orientation** enum);
- Fixed overlaps in TreeLayout when arranging nodes of different sizes;
- Anchor points indices are now serialized;
- Deserialization of custom item classes now works regardless of whether names contain Node or Link suffix;

New in version 2.5

Appearance improvements

- Set the Shape property of tables and containers to RoundedRectangle to render them with rounded corners.
- Hide the frames of table cells by setting the CellFrameStyle property to None.
- Set the EnableStyledText property of TableNode class to render styled text in tables.

New events

- The control raises cellTextEdited event when users edit the text of table cells.
- The createEditControl event lets you create custom DOM element or fragment to use as in-place text editor.
- NodeListView raises nodeSelected event when the user selects a node.

Miscellaneous

- Width and Height properties and the corresponding setter methods for DiagramView, NodeListView, Overview and ZoomControl are now obsolete. You can use the new helper methods of the controls to pass an HTML attributes object, containing style data.
- Stop the controls from adding references to the client scripts by disabling the AddScriptReferences property.
- the loadFromXml method of Diagram class lets you load XML files from client side.
- fixed setZoomFactorPivot bug in virtual scroll mode.

New in version 2.4

Styled text

The EnableStyledText property of ShapeNode allows using HTML-like formatting tags to apply various attributes to the node's text. At this time the component supports the following formatting tags:

- <b> specifies bold text
- <i> specifies italic text
- <u> specifies underlined text
- <color=value> specifies text color
-
 specifies line break

Improved text rendering

- General text rendering quality improved. The component removes scale transformations applied for MeasureUnit and ZoomFactor before drawing text on the canvas, and instead specifies a scaled font size, which helps improve text appearance in Firefox and older versions of Chrome.
- Now items' text can be drawn underlined. To enable this, set the underline attribute of the Font class.
- Font styles can be specified via Style instance by setting its FontStyle property.

License keys

There is no separate trial build of the control assemblies anymore. Instead, set the LicenseKey property to disable the component's evaluation mode and stop displaying trial messages. If your application contains more than one **DiagramView** instance or other controls by MindFusion, you could call **MindFusion.Licensing.LicenseManager.AddLicense(key)** to specify the key once instead of setting it per each control. License key strings are listed on the Keys & Downloads page at MindFusion's customer portal.

Miscellaneous

- Items can be rendered using dashed lines in browsers that support the **setLineDash** function. To enable this, set pen's DashStyle property for DiagramItem or the StrokeDashStyle property of Style.
- Client-side TreeLayout supports organizational charts with assistant nodes as in Microsoft Office diagrams. To mark nodes as assistants, set node.LayoutTraits[TreeLayout.Assistant] to a member of the AssistantType enumeration . Set TreeLayout.enableAssistants = true to arrange assistant nodes in a separate branch between the main node levels.
- Specify if separate connected components of a graph should be arranged horizontally or vertically relatively to each other by setting the multipleGraphsPlacement attribute of layout classes.
- The type of Margin property of LinkLabel has been changed from number to Thickness, letting you specify different margin sizes at each side of the label.
- Masters for Start and Arrow7 shapes added to VisioExporter template file.
- VisioImporter now tries to preserve Shape of ShapeNode objects when importing from files created by **VisioExporter**.

New in version 2.3

Import diagrams from SVG

The SvgImporter class allows importing Scalable Vector Graphics files (*.svg) into MindFusion.Diagramming. Each top-level SVG group element is imported as a separate SvgNode, whose Content is set to the group's child elements. If the SVG file was exported from Visio or OpenOffice Draw, **SvgImporter** will create DiagramLink objects for groups it recognizes as connectors in order to recreate the original diagram model, though in this case it will not preserve fully the original appearance of connectors in SVG.

Zoom control

The ZoomControl class lets users change interactively the current zoom level and scroll position of a Diagram. Bind the **ZoomControl** to the corresponding Diagram by setting its TargetId property. Set the ZoomStep and ScrollStep properties to specify the amount added to diagram's zoom level or scroll position by ZoomControl's buttons. You can customize the control's appearance by setting properties such as Fill, BorderColor, CornerRadius and TickPosition.

Shadow styles

The Diagram.ShadowsStyle property and ShadowsStyle enumeration let you switch shadow rendering order between rendering all shadows in a single background layer (OneLevel) and drawing each shadow close to its node, possibly overlapping other nodes (**ZOrder**). **ShadowsStyle.None** lets you disable shadows altogether.

Miscellaneous

- Set the LinkLabel.Brush property to fill the background of link labels.
- Number of link segments can now be set by calling the DiagramLink.setSegmentCount() method.
- The BackgroundImageAlign property specifies the alignment of diagram's BackgroundImage.
- The TextPadding property specifies the padding distance between the borders of a node and its text.
- Nodes of all classes can be rotated.
- VisioExporter export speed has been improved greatly for large diagrams.
- For consistence with MindFusion libraries for other platforms, BackBrush in default diagram style has been changed to white brush.
- Fixed KeepInsideParent constraint for rotated parent nodes.
- Set the AutoGrow property to **false** to prevent containers from resizing automatically when they would not fit newly added nodes.
- The Brush property of LinkLabel class lets you set background color for link labels.
- Now the NodeModified event is raised after the undo record for that modification has been added to the command history. This makes it easier to merge the ModifyItemCmd with custom commands created from **NodeModified** handler.
- The ChangeUnit method sets a new MeasureUnit and rescales the coordinates of diagram items to keep them the same size.
- OrthogonalLayout no longer ignores repeating links, and will now place them at a distance specified via RepeatingLinksPadding property.

New in version 2.2

SVG Nodes

The SvgNode class represents nodes that can display SVG drawings. SVG graphics are rendered instead of Image, and on top of the geometry rendered by the base ShapeNode class. The SVG drawing of the node is specified through its SvgUrl property.

jQuery support

Now the control can use jQuery for browser abstraction instead of Microsoft Ajax. To choose between Microsoft Ajax and jQuery support use the **ClientSideMode** property.

Visio2013Exporter improvements

- Support for multi-segment links
- Bezier links are now exported as NURBS lines
- Support for rotated nodes
- Proper scale for all MeasureUnit values
- Support for semi-transparent colors

Miscellaneous

- Set the BackgroundImageUrl property to draw an image as Diagram's background.
- The TextStyle property of links provides support for several different text position and orientation styles, defined by the LinkTextStyle enumeration.
- Set the AllowUnconnectedLinks property to true to let users draw links without connecting them to nodes.

- OrthogonalLayout no longer ignores repeating links, and will now place them at a distance specified via RepeatingLinksPadding property.

New in version 2.1

Tree map layout

Tree maps represent hierarchies by nesting child nodes within their parents, where the areas of leaf nodes are proportional to their Weight values. Unlike other layout algorithms, TreeMapLayout expects hierarchies to be defined via grouping or containment (see AttachTo method and ContainerNode class), and will ignore any links in the diagram. The diagram area covered by the topmost nodes in a hierarchy is specified via the LayoutArea property. By default, the layout tries to keep the ratio of node sides as close as possible to one. However this could make it hard to distinguish separate levels of the hierarchy. To alleviate that, set Squarify to **false**, and child nodes will be arranged either as a row or a column inside their parent node, alternating directions for each level. The drawback is that when **Weight** ratios differ greatly or nodes contain many children, some nodes could end up with very narrow rectangles.

Decision flowchart layout

DecisionLayout arranges simple flowcharts consisting of decision boxes with up to three outgoing links per node and activity boxes with a single outgoing link per node. The nodes are arranged in columns and rows, whose distance depends on the HorizontalPadding and VerticalPadding property values. When links share the same row or column, they are placed at a distance specified via LinkPadding. The layout arranges nodes recursively starting from StartNode. If **StartNode** is not specified, the algorithm selects the root of the deepest branch of the graph's spanning tree as start node.

Miscellaneous

- Layout algorithms now automatically resize diagram's Bounds if it's not large enough to fit the arranged content. To revert to old behavior and keep fixed **Bounds** size, set GrowToFit to **false**.
- Fixed hit-testing offset bug when using version 4 of Microsoft Ajax in Chrome browser.
- Fixed error when using gradient brushes for painting swimlane grid cells.

New in version 2.0

Export Visio 2013 files

The Visio2013Exporter class can export the contents of Diagram and DiagramDocument objects to Visio 2013 VSDX files. In order to use it, add a reference to the *MindFusion.Diagramming.Export.Visio.dll* assembly. Similarly to the VDX format exporter, **Visio2013Exporter** relies on a template file, called *Visio2013.vxt*, containing some boilerplate Visio XML code. The template must be either located in same folder as the application assemblies, or its path must be specified via the TemplatePath property. At this time the exporter supports shapes from Visio basic and flowchart stencils.

Visio2013Importer improvements

- The importer now supports Visio group shapes. Child shapes in groups are no longer lost when importing, and their corresponding diagram nodes are attached to the corresponding group node.
- Visio NURBS curves are now imported as Bezier-style links.
- Fix for importing connectors with applied rotation transform.
- Shapes with a 'hidden' attribute in master definition are now imported as invisible items.

Headers and footers in PDF

PdfExporter can now add headers and footers to exported pages. To specify their contents and font, set the HeaderFormat, HeaderFont, FooterFormat and FooterFont properties. Adding a %P placeholder to **HeaderFormat** or **FooterFormat** will render the current page number at its position.

Resize multiple nodes

Now it is possible to resize multiple selected nodes simultaneously. To enable that, set the AllowMultipleResize property to true. When enabled, dragging a corner or side adjustment handle of any node resizes all nodes in the selection. The operation will be cancelled for all nodes if any of them does not accept its new size, e.g. if it's smaller than MinimumNodeSize value. The operation can be cancelled on the client side for all nodes by handling the selectionModifying validation event.

Undo / redo support

If the UndoEnabled property is set to **true**, the Diagram control tracks changes done to its items and allows undoing and redoing them later by calling the undo and redo methods respectively. Multiple changes could be recorded as a single undoable operation by enclosing them between startCompositeOperation and commitCompositeOperation calls. It is also possible to create custom undoable operations by deriving from the Command class and calling executeCommand with the custom command as argument.

Animations

Diagram items can now be animated through the Animation class in the MindFusion.Animations namespace. The class implements several built-in animation and easing types, and allows definition of custom animation functions, to create fluid and pleasing animation effects. The new Animations example demonstrates how to apply several animation types to both nodes and links.

Clipboard support

A single item or a selection of items can be copied or cut to a clipboard maintained by the diagram control. This is done programmatically using the copyToClipboard and cutToClipboard methods. To paste the clipboard contents into the current diagram, call pasteFromClipboard. If running under Internet Explorer, it is also possible to copy to the system clipboard by passing a **true** argument to clipboard methods.

Swimlanes

The lane grid lets you emphasize the relationship between a group of diagram items by displaying them in a distinct lane or cell within the grid. To display the lane grid, set the ShowLaneGrid property of the Diagram class to **true**. In order to customize the grid, set the various attributes exposed by the LaneGrid property, which lets you specify the number of rows and columns, add headers, customize the cell appearance, etc. The new Lanes sample page uses the lane grid to let users draw Gantt charts.

Magnifier

The new magnifier tool allows users to interactively zoom in (or out) portions of the diagram by holding down a modifier key or pressing a mouse button. The magnifier's appearance can be customized via properties such as MagnifierShape and MagnifierFrameColor. Its size and magnification can be set via MagnifierWidth, MagnifierHeight and MagnifierFactor properties. The magnifier appears when users press the mouse button mapped to MouseButtonActions.**Magnify**. It can also be shown programmatically by setting the MagnifierEnabled property.

Intellisense support

The package now includes a MindFusion.Diagramming-vsdoc.js file providing code completion information. To load it in Visual Studio, add e /// <reference path="MindFusion.Diagramming-vsdoc.js" /> tag to the top of your script files. Use the static Diagram.create and Diagram.find methods instead of $create and $find to let Visual Studio infer the type of returned Diagram objects correctly.

Miscellaneous

- Several new **shapes** added for better compatibility with Visio 2013 basic stencil: RightTriangle, Decagon, Trapezoid, Star4Pointed, Star5Pointed, Star6Pointed, Star7Pointed, Star16Pointed, Star24Pointed, Star32Pointed, Donut, Plaque.
- Set the ModificationStart property to AutoHandles to let users start moving or resizing an item without selecting it first.
- The diagram area can be resized automatically to fit the current diagram items as set through the AutoResize property.
- The AutoScroll property enables automatic scrolling when the mouse is dragged near the diagram edges.
- enterInplaceEditMode and leaveInplaceEditMode events raised when the control shows or hides the in-place edit box.
- linkPointed and nodePointed events raised when the mouse hovers over an item.
- The control displays a scaled-down copy of the node being dragged from the NodeListView control.
- Set Behavior to SelectOnly to let users select existing items but not to modify them or draw new ones.
- Added chainable Set* methods for all DiagramView properties.

API changes

- The HitTestHandle method now returns an AdjustmentHandle result instead of **bool** and **ref int** values.
- The type of HitResult argument of the HitTestAdjustmentHandles event has been changed to **AdjustmentHandle**.
- **AnchorUnitType** enumeration has been renamed to UnitType.
- The **SelectionMoved** event has been renamed to SelectionModified, and is also raised if nodes are resized when AllowMultipleResize is enabled. Event handlers now receive SelectionEventArgs parameter instead of **EventArgs**.
- The **SelectionMoving** event has been renamed to SelectionModifying, and is also raised if nodes are resized when **AllowMultipleResize** is enabled. Event handlers now receive SelectionValidationEventArgs parameter instead of **ValidationEventArgs**

New in version 1.8

Rounded links

Links whose Shape is set to Polyline or **Cascading** can be rendered with rounded joints between their segments. To enable that, set the diagram's RoundedLinks property to true. The RoundedLinksRadius property lets you specify the radius of joint arcs.

Link crossings

The component can represent the crossing point of two links either as an arc (bridge) drawn for the link with higher Z index, or as a cut in the link with lower Z index. To enable that, set the diagram's LinkCrossings property to either Arcs or **Cut** member of **LinkCrossings** enumeration. Set CrossingRadius to specify the radius of crossing arcs.

Visio2013Importer improvements

- The Item argument of ImportItem event handlers is now pre-set to a DiagramItem instance created for the imported Visio shape, letting you avoid parsing some standard data such as Bounds or Text. You can either modify this item's properties, or replace it with a new instance of a different type.
- Fixed a crash when a shape master does not have a name attribute (this happens with custom stencils).
- Fixed a crash when theme1.xml is missing from the imported vsdx archive (this happens with files converted from old Visio versions to 2013 format).
- Fixed a parse exception for some non-English locales.
- Fixed a bug where one-segment connectors were imported at wrong positions.

Miscellaneous

- Node rotation was ignored by link routing algorithm; now links are routed around the rotated node's boundaries.
- The routeAllLinks function pulls apart link segments when they would overlap.
- Set Behavior to Pan to only let users pan the view when dragging the mouse.
- The component now raises clicked event for the diagram when there isn't any item at the click location (which would cause nodeClicked or linkClicked event).

New in version 1.7

Import Visio .vsdx files

The Visio2013Importer class can import .vsdx files, created by Visio 2013. In order to use it, you must add a reference to the *MindFusion.Diagramming.Import.Visio.dll* assembly. Various overloads of the Import method can be used to import the file into a

DiagramDocument whose pages correspond to the Visio drawing pages, or into a single Diagram whose content is merged from all imported pages. **Visio2013Importer** supports the basic flowchart and data flow Visio stencils. For each node shape from the .vsdx file, the importer creates a ShapeNode object whose Shape, Text, Brush and Pen properties are set to values corresponding to the imported Visio shape. For each connector shape, the importer creates a DiagramLink object whose Origin and Destination are set to the corresponding imported nodes, and whose ControlPoints reflect the geometry of the connector.

Adjustment handle styles

Appearance of nodes' adjustment handles can be customized by setting the HandlesStyle property. Supported styles include round and square handles, dashed and hatched frames, and some combinations. Apart from appearance, the handles style also controls what part of a node can be dragged to move or resize it. It is also possible to render custom handles by handling the drawAdjustmentHandles event, and implement custom hit-testing by handling the hitTestAdjustmentHandles event.

Dynamic links

A dynamic link automatically orients its end segments so that they point to the target nodes' centers while the link's nodes are dragged around. If a node's AnchorPattern is set, link ends will align to the closest anchor point instead. Positions of the link's end points are also updated when the next-to-last control points are being modified. To enable this, set the Dynamic property to **true**.

ContainerNode enhancements

- Apply layout algorithms to content of a ContainerNode by calling its arrange method.
- Selectively prevent adding or removing child nodes to/from a container by handling the containerChildAdding and containerChildRemoving events.

Miscellaneous

- The Dirty flag is automatically set when a serializable property of an item or the diagram has been modified. It lets you determine where there are any changes that need to be saved.
- The AdjustmentHandlesSize property specifies the size of adjustment handles.
- The ShowDisabledHandles property controls whether disabled adjustment handles are draw on screen.
- Several new predefined shapes available in the Shapes class - RightTriangle, Decagon, Trapezoid, Star4Pointed, Star5Pointed, Star6Pointed, Star7Pointed, Star16Pointed, Star24Pointed, Star32Pointed, Donut and Plaque.

New in version 1.6

Import OpenOffice Draw files

The DrawImporter class can import .odg files, created by the OpenOffice Draw vector graphics editor. In order to use it, you must add a reference to the *MindFusion.Diagramming.Import.Draw.dll* assembly. Various overloads of the Import method can be used to import the OpenOffice drawing into a DiagramDocument whose pages correspond to the Draw pages, or into a single Diagram whose content is merged from all imported pages. **DrawImporter** supports shapes from the General and Flowchart shape palettes in Draw. For each node shape from the .odg file, the importer creates a ShapeNode object whose Shape, Text, Brush and Pen properties are set to values corresponding to the imported OpenOffice shape. For each connector shape, the importer creates a DiagramLink object whose Origin and Destination are set to the corresponding imported nodes, and whose ControlPoints reflect the geometry of the connector.

ContainerNode improvements

- Child nodes are now drawn by their containers; when containers overlap, the children of a lower container cannot appear in front of the upper container anymore.
- The ClipChildren property specifies whether to clip child items to container's boundaries.
- The ZIndex property no longer changes automatically when dropping nodes into a container.
- The Visible property of child items no longer changes when folding or unfolding containers; this lets you keep invisible items inside containers.
- When selection including a container and its child node is dropped into another container, the child node now remains in its original container.

PdfExporter improvements

- Improved handling of clip regions in custom drawing code.
- PdfExporter now honors the target Rectangle property of linear gradients.
- PDF shadings now include all colors from a ColorBlend.
- More precise character widths set when text includes both eastern glyphs and Latin characters.
- More precise widths set for Cyrillic characters.

Miscellaneous

- New overloads of the DiagramView, Overview and NodeListView extension methods have been added. The **initFunction** parameter can be used to set the name of the javascript function, which will contain the initialization code for the corresponding control.
- controlLoaded event is now raised when a control is loaded on the client-side.
- Added support for Diagram.LinkHitDistance, DiagramItem.HyperLink and DiagramNode.EnabledHandles properties.
- Added support for custom node and link types - check the new CustomNodes sample on how to use them.
- Two new measure units - **Centimeter** and **WpfPoint** are now available.

New in version 1.5

Server Side

Miscellaneous

- DiagramView's *Script properties have been modified to chainable methods.

- Use the FromRequest method of the DiagramView for easier recreation of the view's data after postback.
- FallbackContent property allows specifying of a fallback content for the DiagramView.
- NodeSelectedScript, NodeDeselectedScript, NodeDoubleClickedScript, LinkSelectedScript, LinkDeselectedScript and LinkDoubleClickedScript methods have been added to support the corresponding client-side events.

Client Side

Node effects

Two visual effects, represented by GlassEffect and AeroEffect classes, can be applied to nodes. To apply an effect, create an instance of the respective class, set up its properties, then add the instance to the Effects collection of a node (or to Diagram.NodeEffects collection to apply it to all nodes). Effects can be added, removed or modified at any time and this will immediately reflect on the diagram. Effects of different types can be applied simultaneously. It is also possible to apply more than one effect of the same type.

Radial gradients

Nodes can be filled using radial gradients by passing objects in following format to setBrush. If the gradient should blend only two colors, they can be specified using **color1** and **color2** fields instead of **colorStops**.

JavaScript Copy Code

```
var brush =
{
 type: "RadialGradientBrush",
 colorStops:
 [
  { position: 0, color: "white" },
  { position: 0.5, color: "lightGray" },
  { position: 1, color: "blue" }
 ],
 x1: 20, y1: 20, radius1: 30,
 x2: 40, y2: 20, radius2: 80
};
node.setBrush(brush);
```

Multiple labels per link

The LinkLabel class allows multiple captions to be displayed for a single DiagramLink object. Link labels provide a set of properties allowing full customization of their display and positioning. Labels can also be arranged automatically to avoid overlapping nodes and other labels by setting LinkLabel.AutoArrange to true.

New events

- nodeSelected and linkSelected events are now raised when an item is selected, either programmatically or by the user.
- nodeDeselected and linkDeselected events are now raised when an item is deselected, either programmatically or by the user.
- nodeDoubleClicked and linkDoubleClicked events raised when the user double clicks an item.
- nodeCreating is now also raised while the user drags a node from the NodeListView control.

Miscellaneous

- Call setVirtualScroll to enable virtual scrolling mode. In this mode, the canvas stays as big as its containing div, and its content is drawn with a translation bound to the div's scrollbar positions. This allow displaying large diagrams in browsers and on mobile devices where the size of HTML canvas elements is limited.
- resizeToFitText method added to ShapeNode.
- Improved support for nested containers.

New in version 1.4

Server side

Styles and themes

The appearance of the items in a diagram can now be modified using styles. The style represents a set of properties, which can be applied to an individual diagram item (by assigning it to the DiagramItem.Style property) or to all items of a specific type (by using a Theme).
The theme represents a collection of styles. Each style in the theme is associated with the diagram items of a specific type and affects all items of this type. Themes can be loaded from XML files.
For more information about this new feature, check Styles and Themes.

One-way graph Layout

The OneWayLayout class ensures that links enter into nodes from the same general direction and exit them from the opposite side. If the graph contains cycles, some links bend around the nodes to keep the enter/exit direction consistent. The algorithm aims to minimize the number of such links.

Composite layout improvements

The CompositeLayout exposes several new properties. The UseAdditionalPartitioning property can be set to true in order to enable additional partitioning pass after the default partition. This is especially useful when the initial partitioning produces a graph with nodes with many leaf children.

Parallel layouts

The .NET 4 version of **MindFusion.Diagramming** takes advantage of the **Task Parallel Library (TPL)** to enable parallelism on the different layout algorithms. Set the EnableParallelism property on the Layout class to true to enable arranging different subgraphs on different threads to possibly improve performance on multiprocessor systems. In addition, the maximum degree of concurrency can be manipulated through the MaxDegreeOfParallelism property. Note that only independent subgraphs are processed in parallel, so the **SplitGraph** property of layout classes that provide it must also be set.

Preserve order in tree layout

The Balance property of TreeLayout can be set to **Preserve** to keep the original geometric order of child nodes when arranging them under their parent. This can be used to arrange tree branches in a specific order by first positioning them in increasing horizontal or vertical positions.

Export improvements

- RenderOptions property added to PdfExporter, SvgExporter and **WmfExporter** allows controlling various aspects of exported diagrams.
- Speed of PDF generation by **PdfExporter** greatly improved.
- VisioExporter.DynamicsOff can be used to disable dynamic link routing in Visio 2010 in order to preserve the position of control points from the exported diagram;
- **PdfExporter** now creates PDF Bezier objects for Bezier curves in the diagram instead of approximating them with lines;
- If VisioImporter.ApplyVisioUnit is enabled, the diagram's unit will be set to the one from the imported VDX file;

Miscellaneous

- The value of DiagramItem.ZIndex property is no longer required to be unique and smaller than the number of items;
- Diagram item collection types now implement the generic **IList** interface;
- The last row in a TableNode will now be rendered clipped if it fits only partially inside the node, whereas in previous versions the row would not be rendered at all.
- The AllowAddChildren and AllowRemoveChildren properties of ContainerNode specify whether users are allowed to add or remove child nodes to / from the container.
- The GetNearestBorderPoint method of DiagramNode can be used to find a point from the node's border closest to a specified point.

API changes

- RotationAngle property moved from ShapeNode to base DiagramNode;
- Type of Group.FollowMasterRotation changed from boolean to enum GroupRotation;
- **LinkStyle** enum renamed to LinkShape;
- **DiagramLink.Style** property renamed to Shape;
- The type of **TextColor** and **CaptionColor** properties have been changed to Brush; the properties have been renamed to TextBrush and CaptionBrush;
- The default value of **Pen**, **Brush** and **Font** properties of items is now **null** (**Nothing** in Visual Basic) to allow for inheritance from DiagramItem.Style or Diagram.Theme.
- ZIndex no longer corresponds to the position of items within Diagram.Items collection. Items however still contains the items sorted by Z order, as defined by item's **ZIndex**;
- Event handler delegates have been removed; events are now defined as generic EventHandler specialized via EventArgs type. For example the type of NodeCreated is now EventHandler<NodeEventArgs>;
- InitializeNodeScript and InitializeLinkScript properties have been added to support initializeNode and initializeLink client-side events.
- **NodeListView.AddShapeNode** method renamed to **AddNode**.

Client side

Table Nodes

A TableNode displays tabular data, and can be used to represent table schemas in database diagrams or classes in class-hierarchies. A table contains cells arranged in a grid, and every cell can display text and/or image. Cells, columns and rows can be customized in various ways - there are properties available for setting their extents, style, image and text alignment. The number of cells in a table's grid can be programmatically changed by calling methods such as redimTable, insertRow and insertColumn. If a table contains many rows, you can let users scroll them up or down by calling the setScrollable method.

Container Nodes

A ContainerNode can contain other diagram nodes. When a container is moved, its child nodes move too. Containers can be nested one within another to unlimited depth. They can be folded to a smaller size, hiding their child items, and unfolded to show content again. Nodes inside a container can be linked to nodes both inside and outside of it. If a link connects items in the same container, the container is not considered an obstacle by the auto-routing algorithm.

Styles and Themes

An instance of the Style class can be used to set items' appearance properties as a single unit via the setStyle method. In order for style attributes to be used, the respective local appearance properties of a DiagramItem objects must have undefined or null values. If an item's Style does not define a value for some attribute, it is looked up in the styling hierarchy, first in the diagram's Style, next in the current Theme, and finally in the set of default styles defined for each item class.

Shadows

Diagram items can now cast shadows. The shadow color is specified via the setShadowColor method and the offset of a shadow from its item is set via setShadowOffsetX and setShadowOffsetY methods. Shadows aren't rendered if getEffectiveShadowColor returns null or undefined value.

Miscellaneous

- The Locked property can be used to disable interaction with a diagram item.
- Improved rendering speed when changing items' appearance properties.

- Call the setTooltip method to associate tooltip text with an item. The tooltip appears when the mouse pointer hovers over an item for a second.
- Specify the line width of item frames via the setStrokeThickness method.
- initializeNode and initializeLink events are raised when users start drawing a node or a link interactively.
- The NodeListView control can now host table and containers. The **addBox** and **selectBox** methods have been replaced by addNode and selectItem methods.
- The setStartPoint and setEndPoint methods of links provide shortcuts for setting the first and last points in ControlPoints.

New in version 1.3

Attaching and grouping items

MvcDiagram allows attaching a node to another node, establishing a subordinate - master relationship between them. The easiest way to attach nodes is to call the attachTo method. When a master node is moved, all of its subordinates follow it, so that the initial distance between them stays constant. This happens both when a user moves the master node around, and when the node's position is changed using a method or a property.

Collapse and expand tree branches

If a diagram represents a tree structure, the tree branch starting from a node can be collapsed or expanded again in order to hide or display hierarchy details. To enable this, call the setExpandable method. There is a [±] button displayed beside an expandable node to allow users to collapse or expand the tree branch starting there. By default, pressing that button automatically toggles the Expanded property of the node and raises the TreeExpanded or TreeCollapsed events.

User interaction modes

Call setBehavior with one of the Behavior members as argument to specify how the diagram should handle mouse input.

- DoNothing: the diagram is read-only, but nodeClicked and linkClicked events are still raised.
- Modify: existing diagram items can be selected and modified, but no new items can be created interactively.
- DrawLinks: existing nodes can be linked interactively, but no new nodes can be drawn.
- DrawShapes: drawing with the mouse creates new ShapeNode objects; links cannot be created interactively.
- LinkShapes: this is the default behavior, where drawing over an empty spot of the diagram creates a **ShapeNode**, and drawing over a node creates a link.

Touch input

When running under the iOS Safari web browser, the control handles one-finger touch events to allow drawing and selecting items interactively. In addition, the hit-test area around adjustment handles is larger to make touch modifications easier. Two-finger gestures are delegated to the browser - they either scroll the canvas when dragging in the same direction or zoom the page when dragging in opposite directions. The preventDefaultTouch flag can be set to false to revert to the default one-finger touch behavior (scroll the whole page).

Miscellaneous

- The Visible property specifies whether an item should be shown on screen.
- Validation events added: NodeCreating, LinkCreating, NodeDeleting, LinkDeleting, NodeModifying and LinkModifying.
- The ScrollX and ScrollY properties and scrollTo method lets you set the diagram's scroll position.
- getViewport returns the currently visible part of the diagram.
- setZoomFactorPivot sets the zoom level in such way that the pivot point's screen position does not change.
- zoomToRect zooms and scrolls the view to fit the specified document rectangle in the Diagram's visible area.
- zoomToFit zooms the view to fit the document contents into the current view size.
- getContentBounds returns the smallest rectangle that bounds all diagram objects.
- keepGroupLayout indicates whether to treat each group of attached nodes as a single vertex in the arranged graph.
- Extension methods are modified to add MVC3 Razor support.

New in version 1.2

Server side

Fractal tree Layout

FractalLayout is a tree layout algorithm that places child nodes symmetrically around their parent node. Nodes at the lowest level are arranged directly in a circle around their parent. At upper levels, the already arranged nodes form branches that are arranged in a circle around the new parent node. The algorithm is recursively repeated till the highest level is reached. If nodes in the tree have uniform number of children, the end result has fractal-like appearance (subsets of the graph look like scaled-down copies of the whole graph). You can choose which node should be displayed at the center of the topmost circle by setting the Root property. If it is not specified, the algorithm automatically selects a root that leads to more balanced distribution of nodes.

LayeredLayout enhancements

- Specify start and end nodes in LayeredLayout via the StartNode and EndNode properties.
- EnforceLinkFlow specifies whether to attempt to force all links to go in the same direction. If the graph being arranged contains cycles, some of the links will still go backward.

Client side

Alignment grid

The alignment grid help users place the diagram items more precisely. To activate the grid, set the AlignToGrid property to true. If the grid is active while an item is being created or modified, the item's control-points are aligned to the nearest grid points. The distance between adjacent grid points is set via the GridSizeX and GridSizeY properties. The grid can be either visible or invisible,

depending on whether the ShowGrid value is true or false. If visible, the grid is painted as a matrix of points or as series of crossing lines, as specified by the GridStyle property. GridColor defines the color with which the alignment points or lines are painted.

Bordered tree layout

In contrast to **TreeLayout**, which centers the parent node above its child nodes, BorderedTreeLayout arranges children in columns at a horizontal offset form their parent. **BorderedTreeLayout** members control the tree direction, how much space to leave between tree levels, and between nodes on the same level.

Anchor points

The AnchorPattern class represents a set of AnchorPoint objects, which specify the exact locations where links are allowed to connect to nodes. A pattern can be assigned to nodes by calling their setAnchorPattern method. The **AnchorPattern** class exposes several predefined patterns as static fields: decision2In2Out, decision1In3Out, leftInRightOut, topInBottomOut.

MvcDiagram Fundamentals

MvcDiagram consists of two main components, a .NET class library running on the server and a client side control implemented in JavaScript.

Server side

The .NET library implements a flow-diagramming object model that lets you define structures such as trees, flowcharts and graphs. The most important classes from the library are Diagram, ShapeNode, TableNode, ContainerNode, SvgNode and DiagramLink. A **Diagram** instance serves as the container for relational graph data, whose content is accessed through the Nodes and Links collections. **Nodes** contain DiagramNode instances that represent the nodes (a.k.a. vertices) in a graph or tree, and **Links** contain **DiagramLink** instances that represents the links (a.k.a. edges/arcs) between nodes in the graph. Usually a **Diagram** instance will be a part of a Model class in an MVC application, or will be used as an intermediate object that only maps the model data to the client-side diagram view.

Once the **Diagram** is populated and DiagramView configured to draw it and specify interaction mode, it can be rendered as part of an HTML page by calling the DiagramView extension method from the MVC view code.

There are also various export classes you can use on the server to export diagrams to PDF, SVG, Flash, DXF, Visio and WMF formats.

Client side

On the client side, a diagram model is represented by the Diagram JavaScript class, and rendered by the DiagramView control that draws the diagram content on an HTML Canvas. The view also allows users to add new nodes and links to the diagram by drawing them with the mouse. New nodes can also be created via drag-and-drop from the NodeListView control.

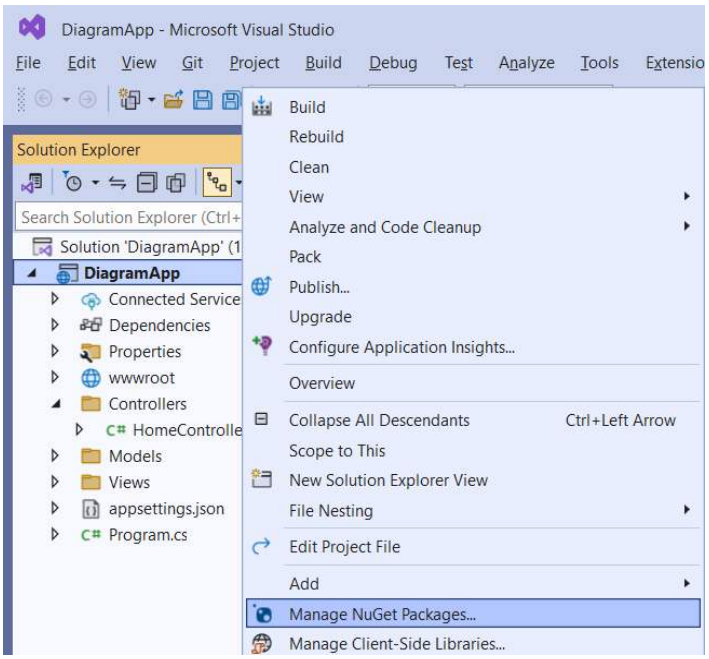
Some of the automatic layout features you might know from other MindFusion diagramming controls are implemented on the client side, including Layered, Tree, Spring and Fractal layouts.

Getting Started

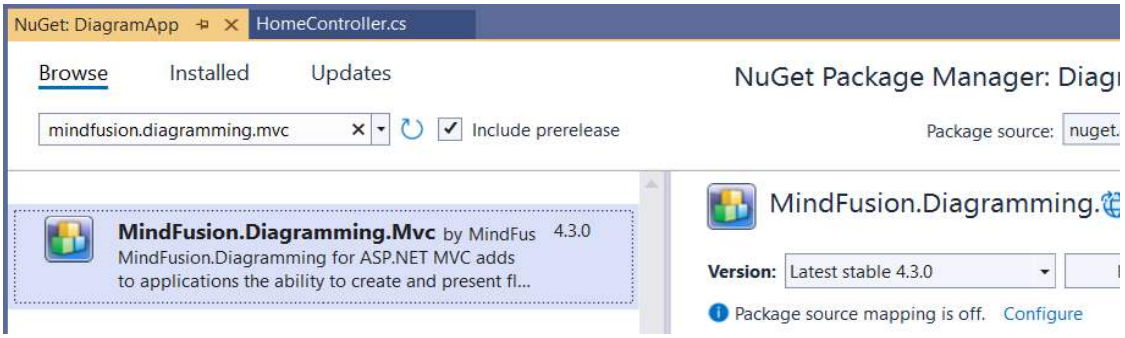
The following topics describe how to integrate the Diagramming API into ASP.NET Core Web App projects (either MVC or Razor Pages), and create an instance of the DiagramView control.

Add Nuget Package

In order to integrate the Diagramming API into your ASP.NET Core Web App (either MVC or Razor Pages), select Manage Nuget Packages from Project menu:



Open the Browse tab, and enter *MindFusion.Diagramming.Mvc* in the search field. Select the *MindFusion.Diagramming.Mvc* package and click Install.



Once the package is installed, you should be able to add components from the library to your UI (DiagramView, NodeListView, etc).

Create DiagramView in MVC Project

After following the previous topic, you should be able to add a DiagramView component to your ASP.NET MVC (Model-View-Controller) user interface.

1. Create a DiagramView component

In *HomeController.cs* add a reference to the MindFusion.Diagramming.Mvc namespace:

C# Copy Code

```
using MindFusion.Diagramming.Mvc;
```

In the body of the **Index** method, create a DiagramView instance and store it in the **ViewBag**:

C# Copy Code

```
DiagramView view = new DiagramView("diagramView1");
ViewBag.DiagramView = view;
```

2. Render DiagramView inside a view

In *Index.cshtml* add a reference to the MindFusion.Diagramming.Mvc namespace:

Razor Copy Code

```
@using MindFusion.Diagramming.Mvc
```

If you want to use tag helpers for adding MindFusion components to views, import them with the @addTagHelper directive:

Razor Copy Code

```
@addTagHelper *, MindFusion.Diagramming.Mvc
```

Add the component by using either the DiagramView HTML helper:

Razor Copy Code

```
@Html.DiagramView(
    (DiagramView)ViewBag.DiagramView,
    new { style = "width:100%; height:100%;" })
```

or alternatively using its Tag helper:

Razor Copy Code

```
<mindfusion-diagramview
    component=(DiagramView)ViewBag.DiagramView
    style="width:100%; height:100%;">
</mindfusion-diagramview>
```

3. Create diagram items

You can create nodes and links programmatically either by calling Factory methods, or by instantiating objects of respective class and adding them to Items collection:

C# Copy Code

```
// in controller

DiagramView view = new DiagramView("diagramView1");

var df = view.Diagram.Factory;

var node1 = df.CreateShapeNode(10, 10, 30, 30);
node1.Text = "Hello";

var node2 = df.CreateShapeNode(60, 25, 30, 30);
node2.Text = "ASP.NET MVC";

df.CreateDiagramLink(node1, node2);

ViewBag.DiagramView = view;
```

4. Preserve data between postbacks

The DiagramView component can preserve its data between postbacks by serializing and deserializing to JSON internally. Call the static FromRequest method
to get this functionality. You can check the request's content type to handle the initial setup of the view and its diagram.

C# Copy Code

```
// read from request
DiagramView view = DiagramView.FromRequest("diagramView1", Request);

// initial setup
if (!Request.HasFormContentType)
{
     var df = view.Diagram.Factory;

     var node1 = df.CreateShapeNode(10, 10, 30, 30);
     node1.Text = "Hello";

     var node2 = df.CreateShapeNode(60, 25, 30, 30);
     node2.Text = "ASP.NET MVC";

     df.CreateDiagramLink(node1, node2);
}

ViewBag.DiagramView = view;
```

Browse the other sections of this help file to learn more about the functionality provided by the MindFusion.Diagramming classes, methods and properties. Follow the tutorials to learn how to populate the diagram model programmatically or create custom item types.

Create DiagramView in Razor Pages Project

After adding the nuget package, you should be able to add a DiagramView component to your ASP.NET Razor Pages user interface.

1. Create a DiagramView component

In *Index.cshtml.cs*, add a reference to the MindFusion.Diagramming.Mvc namespace:

C# Copy Code

```
using MindFusion.Diagramming.Mvc;
```

In the body of the **Index** method, create a DiagramView instance and store it in the **ViewData**:

C# Copy Code

```
DiagramView view = new DiagramView("diagramView1");
ViewData["DiagramView"] = view;
```

2. Render DiagramView inside a view

In *Index.cshtml*, add a reference to the MindFusion.Diagramming.Mvc namespace:

Razor Copy Code

```
@using MindFusion.Diagramming.Mvc
```

If you want to use tag helpers for adding MindFusion components to views, import them with the @addTagHelper directive:

Razor Copy Code

```
@addTagHelper *, MindFusion.Diagramming.Mvc
```

Add the component by using either the DiagramView HTML helper:

Razor Copy Code

```
@Html.DiagramView(
    (DiagramView)ViewBag.DiagramView,
    new { style = "width:100%; height:100%;" })
```

or alternatively using its Tag helper:

Razor Copy Code

```
<mindfusion-diagramview
    component=(DiagramView)ViewBag.DiagramView
    style="width:100%; height:100%;">
</mindfusion-diagramview>
```

3. Create diagram items

You can create nodes and links programmatically either by calling Factory methods, or by instantiating objects of respective class and adding them to Items collection:

C# Copy Code

```
// in controller

DiagramView view = new DiagramView("diagramView1");

var df = view.Diagram.Factory;

var node1 = df.CreateShapeNode(10, 10, 30, 30);
node1.Text = "Hello";

var node2 = df.CreateShapeNode(60, 25, 30, 30);
node2.Text = "ASP.NET Razor Pages";

df.CreateDiagramLink(node1, node2);

ViewData["DiagramView"] = view;
```

4. Preserve data between postbacks

The DiagramView component can preserve its data between postbacks by serializing and deserializing to JSON internally. Call the static FromRequest method from the page's **OnPost** method to get this functionality.

C# Copy Code

```
public void OnPost()
{
    DiagramView view = DiagramView.FromRequest("diagramView1", Request);
    ViewData["DiagramView"] = view;
}
```

Browse the other sections of this help file to learn more about the functionality provided by the MindFusion.Diagramming classes, methods and properties. Follow the tutorials to learn how to populate the diagram model programmatically or create custom item types.

User Interaction

Most likely an application will use MvcDiagram to enable users to interactively create diagrams, flowcharts and other kinds of presentation documents. A user composes or edits a diagram by drawing its elements on the diagram canvas area. With the help of a mouse or another pointing device users draw shapes and links, possibly connecting them in complex hierarchies, graphs, trees or other structures they might need.

Creating New Items

New items can be created by drawing them with the mouse. While a user drags the mouse, the view of the control is updated to show what the new item would look like at the pointer location. If the operation cannot be completed at the current mouse position, the cursor changes to a crossed out circle. When the user releases the button, the item is created. The control does not let users create very tiny nodes which would be hard to select and modify. By default links can be drawn only between nodes, unless the AllowUnconnectedLinks property is enabled. Users can cancel item creation by pressing the right mouse button while drawing.

A node can also be created via drag-and-drop from the NodeListView control. Its instance can be added to the html page by calling the NodeListView extension method and specifying a collection of prototype nodes that should be listed in the palette.

Modifying Items

A user modifies an item by dragging its manipulation handles. The **ModificationStart** property specifies when these handles appear. The first option, AutoHandles, makes the handles appear when the mouse hovers over an item; modification of the item can start immediately by pointing it and dragging a handle. The second option is **SelectedOnly**: in order to be modified, an item should first be selected. Manipulation handles appear around the item to indicate that now it can be moved or resized. Number and functionality of those handles depend on the item type and the value of the corresponding HandlesStyle property.

Modifying a Node

Usually nodes have nine adjustment handles. The one at the center can be used to move the node. When dragged, each corner handle resizes the node by moving the two edges adjacent to the corresponding corner. The edge handles placed at the center of each side can be used to resize a node either horizontally or vertically. Shape nodes can be rotated using an additional round handle drawn above them. Each handle can be enabled or disabled by setting or clearing the respective bit in EnabledHandles.

Modifying a Link

Variety and behavior of link adjustment handles depend on the link's Shape and SegmentCount. Control-points stay at both ends of each segment that constitutes a link. MvcDiagram offers three link styles. Following sections discuss how to modify links of each style.

- The first style called Polyline defines links, which consist of a series of straight segments, drawn between control-points. There are no restrictions as to the number of segments. Dragging a control-point modifies each segment adjacent to the point. If the AllowSplitLinks property is enabled, users are allowed to insert new control points or to remove existing ones. Inserting a control point is done by pressing the left mouse button over a segment and dragging. The new point appears at the mouse location and splits the pointed segment in two.
- If link's style is set to **Bezier**, each segment has 4 manipulation handles that define a curve. Each curve goes through the end-points, which are shaped as circles. Bend of the curve is defined by the middle two points shaped as rectangles. If a user drags an end-point modification handle, the adjacent control points will be moved with same translation vector as that of the end-point. If a bend control point is dragged, the symmetric control point of the neighboring curve is also moved to ensure the two stay on the same line with the end-point and on the same distance from it.
- **Cascading** links consist of straight horizontal and vertical segments, each one orthogonal to its neighbors. If a control-point is dragged, the segments adjacent to it are moved and resized preserving their orthogonal orientation to each other. If the **AllowSplitLinks** property is enabled, users are allowed to insert new control points to **Cascading** links or remove existing ones. The orientation of the link segments can be changed if CascadeOrientation is set to Auto.

Selecting Items

Multiple items are selected by dragging while CTRL key is down. Selection can also be toggled with CTRL + clicking an item. A rectangle appears around selected items, allowing a user to move all of them at once.

Multi-touch Support

- If MultiTouchZoom property is enabled (default), the view can be zoomed or panned using two-touch pinch / flick gestures.
- If MultiTouchModify property is enabled (default), diagram nodes can be moved, scaled and rotated using two-touch pinch / flick gestures.
- If **MultiTouchZoom** property is disabled, each touch draws diagram items corresponding to current behavior.
- If **MultiTouchModify** property is disabled, each touch started from a node draws a diagram link.

Latter modes can be used for collaborative whiteboarding / classroom scenarios.

Programming Interface Overview

Model-View Architecture

Apart from targeting the ASP.NET MVC framework, MvcDiagram also implements the Model-View-Controller (MVC) architectural pattern internally. As applied to MindFusion.Diagramming classes, it separates the diagram data (model) from its representation to the user (view) and controlling it by the user (controller).

Single Diagram Canvas

The Diagram class represents the diagram data model and provides operations to process the data. It exposes properties and methods for adding new diagram elements, finding and accessing existing elements, or removing them.

The DiagramView class presents the relevant data from the **Diagram** class by drawing on the screen using the **CanvasRenderingContext2D** API. When the diagram changes, the view automatically redraws the affected part of the image to reflect the changes. Multiple views can display the same diagram, set by their Diagram property. Each view can have a different zoom level and scroll position.

The **DiagramView** class creates Controller objects when user starts interacting with the diagram, depending on the value of view's Behavior property. That property specifies how GUI actions to manipulate the data should be interpreted.

Multiple Diagram Pages

The DiagramDocument class represents a collection of diagram pages or sheets, represented by DiagramPage objects. New pages can be added to the document and existing pages can be removed and reordered by modifying the Pages collection.
The SaveToJsonFile, SaveToXml and SaveToString methods of **DiagramDocument** save all pages in a single file, and respectively the LoadFromJsonFile, LoadFromXml and LoadFromString methods load all pages from the file.

DiagramPage is derived from Diagram and only adds a few properties, so any code that processes **Diagram** objects will work with **DiagramPage** objects too. A diagram document forwards each event raised by its pages, giving you a chance to use the same handler for events raised by different pages. If a page should implement distinct user interaction or validation logic, it is also possible to attach handlers to the **Diagram** events inherited by **DiagramPage**.

A single **DiagramPage** could be assigned to the Diagram property of DiagramView, and that lets you create your own user interface for presenting a list of pages to the user and selecting a current page. The package also includes a TabbedDiagramView that presents all pages of a document in a tabbed user interface where each tab corresponds to a page. The page displayed in the active tab is exposed by the selectedPage property of TabbedDiagramView. The active page can be changed interactively by activating its associated tab. New pages can be added and removed to / from the document through buttons in the tab strip and the pages can be rearranged by dragging their associated tabs with the mouse.

User Interface Components

The following UI components from the library can be added to a page to let users create and view diagrams:

DiagramView

DiagramView renders the contents of a Diagram, and lets users draw new items by drawing with mouse or touch on the canvas. The Behavior property indicates how the control should interpret users' actions. Additional properties allow scrolling and zooming the view.

TabbedDiagramView

TabbedDiagramView displays multi-page DiagramDocuments. Each DiagramPage is displayed in its own tab. Apart from base DiagramView members, the control includes properties for customizing the tab strip, and to specify whether users are allowed add or remove pages.

Palette

The Palette control represents a tool palette that displays DiagramItem objects grouped into categories, and is implemented as accordion UI. Call the AddCategory method to define a category and create its respective accordion pane (an ItemListView). Call AddItem to add an item to specified category. **Palette** provides same layout and appearance properties as the **ItemListView** class.

ItemListView

ItemListView represents a tool palette that contains uncategorized list of diagram items. Items can be added to the list by calling AddItem method, and removed by calling RemoveItem. Drag-and-drop operations from **ItemListView** to DiagramView create a copy of the dragged item, and set its size to NewInstanceSize.

Overview

Overview provides a scaled-down view of a target DiagramView, helping users work with larger diagrams. A tracker rectangle shows current viewport of the main **DiagramView**, and by dragging it users can change the scroll position. It can also be resized to zoom the target view.

ZoomControl

ZoomControl lets users change the zoom level of a target DiagramView by dragging a slider. It also includes arrow buttons for scrolling the view.

Ruler

The Ruler control displays horizontal and vertical scales that let users measure and align diagram nodes.

Diagram Structure

Standard Item Types

Diagram elements in MvcDiagram are represented by ShapeNode, TableNode, ContainerNode and DiagramLink instances. Combined and linked together, they can form complex data structures such as process or entity-relationship diagrams, class hierarchies, networks and graphs. Each of their rich features can be customized through numerous methods and properties.

Shape Nodes

ShapeNode objects can represent geometric shapes, flowchart elements, icons in network diagrams, and so on. They can display both graphics and text in different fonts, sizes and colors. There are more than 100 predefined node shapes (you can preview them here) and apart from them, MvcDiagram offers an easy way to define custom shapes. Shape nodes can be rotated, either interactively using the round selection handle drawn above them, or programmatically using the RotationAngle property. Each node can also display a distinct bitmap image specified via its ImageUrl property.

Table Nodes

A TableNode displays tabular data, and can be used to represent table schemas in database diagrams or classes in class-hierarchies. A table contains cells arranged in a grid, and every cell can display text and/or image. Cells, columns and rows can be customized in various ways - there are properties available for setting their extents, style, image and text alignment. Tables' dimensions can be programmatically changed; inserting, appending or deleting rows and columns can be done in a very comprehensive, intuitive way whenever you might need it. Tables can also display background images and graphics.

Container Nodes

A ContainerNode can contain other diagram nodes. When a container is moved, its child nodes move too. Containers can be nested one within another to unlimited depth. They can be folded to a smaller size, hiding their child items, and unfolded to show content again. Nodes inside a container can be linked to nodes both inside and outside of it. If a link connects items in the same container, the container is not considered an obstacle by the auto-routing algorithm.

Diagram elements in MvcDiagram are represented by ShapeNode, TableNode, **ControlNode** and DiagramLink instances. Combined and linked together, they can form complex data structures such as process or entity-relationship diagrams, class hierarchies, networks and graphs. Each of their rich features can be customized through numerous methods and properties. MvcDiagram also offers intuitive user interface for drawing and editing diagram items interactively.

SVG Nodes

The SvgNode class represents nodes that can display SVG drawings. SVG graphics are rendered instead of Image, and on top of the geometry rendered by the base ShapeNode class. The SVG drawing of the node is specified through its SvgUrl property.

Composite Nodes

Diagram nodes can be composed from components. Components are graphical widgets similar to controls, which can be arranged in hierarchies and added to a Diagram through instances of the CompositeNode class. The component system contains various built-in components, such as Text and Image. Component hierarchies can be defined and loaded through Json templates using loadTemplate and classFromTemplate methods.

Links

DiagramLink objects are used to represent various types of connections and relations in diagrams. MvcDiagram offers three basic link types:

- polyline links composed of straight-line segments;
- curved links composed of Bézier-spline segments;
- links composed of horizontal and vertical segments, alternating their orientation so that every segment is orthogonal to its neighbor.

A link can be connected to a node at each of the link's ends. It is also possible to have unconnected links - having one or both of their ends not connected to nodes.

Custom Composite Nodes

The CompositeNode class makes it easier to add custom graphics primitives to nodes. Appearance of composite nodes is defined by a tree of layout panels and child components. Unlike MindFusion's desktop diagram libraries, node templates are specified using JSON notation instead of XML. A sample client-side node class derived from **CompositeNode** can now be created declaratively using template as below.

C# Copy Code

```
@{
string template = @"{
 ""component"": ""GridPanel"",
 ""rowDefinitions"": [""*""],
 ""columnDefinitions"": [""22"", ""*""],
 ""children"":
 [
  {
   ""component"": ""Rect"",
   ""name"": ""Background"",
   ""pen"": ""black"",
   ""brush"": ""white"",
   ""columnSpan"": 2
  },
  {
   ""component"": ""Image"",
   ""name"": ""Image"",
   ""autoProperty"": true,
   ""location"": ""Images/icon4.png"",
   ""margin"": ""1"",
   ""imageAlign"": ""Fit""
  },
  {
   ""component"": ""StackPanel"",
   ""orientation"": ""Vertical"",
   ""gridColumn"": 1,
   ""margin"": ""1"",
   ""verticalAlignment"": ""Near"",
   ""children"":
   [
    {
     ""component"": ""Text"",
     ""name"": ""Title"",
     ""autoProperty"": true,
```

```
     ""text"": ""title"",
     ""font"": ""Arial bold""
    },
    {
     ""component"": ""Text"",
     ""name"": ""FullName"",
     ""autoProperty"": true,
     ""text"": ""full name"",
     ""pen"": ""blue"",
     ""padding"": ""1,0,1,0""
    },
    {
     ""component"": ""Text"",
     ""name"": ""Details"",
     ""autoProperty"": true,
     ""text"": ""details"",
     ""font"": ""Arial 3""
    }
   ]
  }
 ]
}
";
```

Objects from the MindFusion.Drawing namespace double as **CompositeNode** components. Apart from JSON templates, the visual tree can also be created by adding panels and drawing objects programmatically from JavaScript, starting from the node's root field.

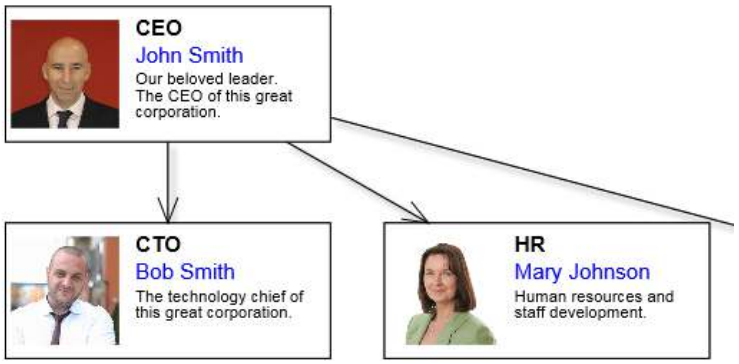
Panel classes include:

- SimplePanel, laying out its children to overlap each other, taking up all available space (useful for background graphics or images).
- StackPanel, laying out its children side by side, either horizontally or vertically.
- GridPanel, laying out its children in rows and columns, where children in same cell overlap. Rows and columns can be either set to a fixed size, auto-size to their child preferred size, or take up all space remaining unoccupied by other rows and columns.

Named components (with specified name value) can be accessed by calling the getComponent method. Components whose autoProperty attribute is enabled will have a pair of get/set property methods generated automatically by classFromTemplate method. The auto-properties from above template can be used as below.

JavaScript Copy Code

```
var node1 = new OrgChartNode(diagram);
node1.setBounds(new Rect(25, 15, 60, 25));
node1.setTitle("CEO");
node1.setFullName("John Smith");
node1.setDetails(
    "Our beloved leader. \r\n" +
    "The CEO of this great corporation.");
node1.setImage("ceo.png");
diagram.addItem(node1);
```



Auto properties are also automatically serialized in JSON format, cloned by node.clone() method and saved/restored by undo commands.

The Shape class used to specify ShapeNode geometry can also be used as a component. When its **isOutline** attribute is set, the shape will control **CompositeNode**'s geometry too, defining hit test and clip area, and link alignment points along node's border. If **isOutline** is disabled, the shape will serve mostly as a decorative element. The code below shows a sample fragment specifying shape component in template:

JavaScript Copy Code

```
{
    component: "Shape",
    id: "Cloud",
    autoProperty: true,
    name: "OutlineShape",
    pen: "gray",
    brush: "Transparent",
    isOutline: true
},
```

If custom nodes will be posted back to server, you must still declare a matching node class as a data container for the custom nodes:

C# Copy Code

```
using MindFusion.Diagramming;
using MindFusion.Diagramming.Mvc;

public class OrgChartNode : CompositeNode
{
    public OrgChartNode()
    {
    }

    public OrgChartNode(Diagram diagram)
        : base(diagram)
    {
    }

    [AutoJson]
    public string Image { get; set; }

    [AutoJson]
    public string Title { get; set; }

    [AutoJson]
    public string FullName { get; set; }

    [AutoJson]
    public string Details { get; set; }
}
```

but unlike older methods of creating custom nodes, you no longer have to create client-script code. The latter is generated automatically from the template and bound to server class properties using a single call to RegisterItemType:

C# Copy Code

```
// register custom item type
view.RegisterItemType(typeof(OrgChartNode), "OrgChartNode", "OrgChartNode", 100, template);
view.LoadFromJson(Request.Form["diagramView1_PostData"]);
ViewBag.DiagramView = view;
```

This is demonstrated in Tutorial 8 sample project installed with the component.

Creating and Deleting Items

Creating Items Programmatically

In order to add a new node or link to a diagram, create an instance of the item class via the new operator and add it to the diagram's Nodes or Links collection. This works the same way for adding both items from your custom class or ones from the MvcDiagram predefined types.
There are several shortcut methods provided for creating instances of the predefined node and link types. The methods are members of the diagram's Factory object:

- The CreateShapeNode method creates a ShapeNode at the specified position.
- There are several overloads of the CreateDiagramLink method that allow connecting two nodes of the same type or of different types.

Drawing New Items

Users can draw diagram elements with the mouse. This creates either a node or a link, depending on whether there is already another diagram item under the mouse cursor. When a user starts drawing, an item is created and its appearance is initialized from the default-value properties of the Diagram, such as ShapePen and ShapeBrush. When the user releases the mouse button, the item is added to the collection that contains items from the same type - either Nodes or Links.

Deleting Items

Call the Remove method of the Nodes or Links collection to delete an item from the diagram. The ClearAll method clears diagram contents by deleting all items.

Accessing Items

One of the most common tasks when using the component is iterating over all items in the diagram document or navigating through the connections between items - from a node to the incoming or outgoing links, or from a link to its origin and destination nodes. To enumerate items of a diagram use the Items, Links or Nodes collections. The nodes that a link connects are referenced by the link's Destination and Origin properties. To access the links that are connected to a node, use its IncomingLinks and OutgoingLinks collections. The first collection contains references to all links pointing to the node, the second - to the links that start from it. Links can connect to rows of a table; to find such links use table's Rows collection and the IncomingLinks and OutgoingLinks collections. A shortcut for accessing the other end of a relation in entity-relationship diagrams is provided by the RelatedOneToMany and RelatedManyToOne properties of the Row class.

Selected Items

If only one item is selected, it is considered the active item and can be accessed with the ActiveItem property. The type of this property is DiagramItem; you can get the actual type of the selected item through .NET reflection. Multiple items can be selected when clicking or dragging with the CTRL key pressed at the same time. Multiple selection is represented by the Selection property of the Diagram class. The Selection object provides collections to access the selected Links and Nodes.

Finding Items

You can search for diagram items by their Tag or Id using the FindLink, FindNode, FindLinkById and FindNodeById methods. Tag and id objects are compared by calling the **Equals** method of the **System.Object** class. Thus if they are of value type, they are compared by value (that is bitwise equality). If they are of reference type their **Equals** method is used. If there are more items with the same tag, the first one will be returned. If no item with the specified tag or id exists, a **null** reference (**Nothing** in Visual Basic) will be returned. Groups can also be searched by tag using FindGroup.

Relationship Queries

The Query method traverses the graph by following node/link connections and group's parent/child relationships to find items in specific relation to the current one. The method receives a string argument consisting of selectors and optional predicates. The query is executed in several steps. In each step the result set is constructed from the previous result set by following the relationship specified by the current selector. If a predicate is specified after the selector identifier, the result set is filtered to include only items for which the predicate evaluates to **true**.

Attaching and Grouping Items

MvcDiagram allows attaching a node to another diagram element, establishing a subordinate - master relationship between the two items. The easiest way to attach nodes is to call the AttachTo method. When a master element is moved or resized, any subordinated node follows it, so that the initial distance between the two items stays constant. This happens both when a user moves the master item around, and when the item position is changed using a method or a property.

> **Note**
>
> Recent versions of the API include NodeLabel and LinkLabel classes, which address several of the scenarios described below. We recommend using these classes when appropriate, as they offer a more lightweight and easily configurable alternative to attached nodes.

Possible Uses

- Attach a Transparent shape node to another item and set its Text, so that the shape node acts as an additional label assigned to the main item.
- Attach a shape node to another item and set the node's Image, so it acts as an additional icon displayed inside or near the main item.
- In all previous cases, the Locked property can be set for attached nodes, which are used only as decorations embellishing the main item. That prevents users from selecting, modifying or drawing links to the subordinates.

Hierarchical Grouping

A unit of attached items is represented by an instance of the Group class. To create a group, instantiate a **Group** object and add it to the diagram's Groups collection, or call the CreateGroup method passing the master object as argument. Attach items to the group

using the various **Group** methods. The AttachTo method of nodes is a shortcut to creating a group and attaching an item to it.

Every member of a group can be the master item of another group, that is, an item can be subordinate in one group and meanwhile be another group's master. MvcDiagram doesn't impose any limits to the depth and width of group hierarchies built that way. However, it is not possible to put an item as a subordinate in more than one groups because the hierarchy relations would become ambiguous.

Unions

It is possible to create such groups, that if any item in the hierarchy is modified, all other items are updated too. This can be done by putting the items in circular chain of dependencies - that is, item1 is attached to item2, item2 to item3 ... itemN to item1. If any of these items is modified, the change will propagate through the chain until it reaches its originator.

Attaching Items to a Group

Links cannot be attached to groups but can be master items of groups. If a group's master item is a link, nodes can be attached to the control points or segments of the link. See the AttachToLinkPoint and AttachToLinkSegment methods for details. If the master item is a node, other nodes can be attached to any of its corners, or to an area specified as percent of master-node's bounding rectangle. Check AttachToCorner and AttachProportional for details. A shortcut to creating Group instance and calling the appropriate AttachTo* methods is provided by the AttachTo method of the DiagramNode class.

Accessing Group Items

Group members can be accessed using the MainItem property and the AttachedNodes collection. To get the groups in which an item participates, use its MasterGroup and SubordinateGroup properties.

Destroying Groups

If a user deletes the master item of a group, the group is destroyed automatically and the GroupDestroyed event is raised. If AutoDeleteItems is enabled, subordinated items in the group are deleted too. If **AutoDeleteItems** is disabled, subordinated items are just detached from their master and remain intact. Groups can be destroyed programmatically by removing them from the diagram's Groups collection.

Laying Out Diagrams Automatically

Applications that process hierarchical or more complex entity-relationship data might need to present that data to end-users in an aesthetical and visually appealing manner. Such applications would store their data as graph of connected nodes possibly having additional information attached to them. However, position and size coordinates of nodes might be unnecessary and/or unavailable. MvcDiagram provides a way to automatically arrange the nodes of a diagram, applying attractive layouts on the overall structure of the diagram. To utilize that functionality, an application can create a diagram by assigning generic coordinates to its nodes, instantiating a layout object and invoking the layout's Arrange method. Non-blocking ArrangeAsync method is available in the .NET 4.5 build of *diagramming.dll* assembly. Arrange methods can be prevented from changing positions of individual items by setting the IgnoreLayout property of those items to **true**.

The classes that implement automatic layout algorithms are located in the MindFusion.Diagramming.Layout namespace. The following sections list the available algorithms categorized by their general implementation strategy: **force-directed algorithms**, **graph topology based algorithms** and **semantics based algorithms**.

Diagram Serialization and Output

Serializing the Diagram

JSON Serialization

To serialize a diagram as a JSON string, call its SaveToJson method. To deserialize a diagram back from its JSON representation, call the LoadFromJson method. The SaveToJsonFile and LoadFromJsonFile methods serialize the JSON strings to / from files.

Custom item classes can implement JSON serialization for their additional properties by overriding the SaveToJson and LoadFromJson methods. Such classes must be registered for serialization by calling the RegisterItemClass method and providing a string identifier for the *clientClass* parameter.

The control raises JsonSerializeTag and JsonDeserializeTag to enable serialization of complex Id and Tag objects without overriding the item classes. The PropertyName event argument specifies which property is being serialized.

JSON serialization is handled internally by the classes from MindFusion.Json namespace. The JsonValue class represents a property value. Property names and values are stored in instances of the JsonObject dictionary class. The JsonContext class handles the serialization of the Diagram properties. The Serialize method is a wrapper over the Serializer class' Serialize method. It takes a

JsonObject and serializes it to a string. The Parse method is a wrapper over the Parser class' ReadJsonObject method. It takes a JSON-formatted string and returns a **JsonObject** with the parsed data.

XML Serialization

It is possible to save a diagram as an XML document. To do that, call the SaveToXml method of the Diagram class. To deserialize a diagram back from its XML representation, call the LoadFromXml method.

Custom item classes can implement XML serialization for their additional properties by overriding the SaveToXml and LoadFromXml methods. In addition, such classes must specify a string to use as their identifier in the XML document by calling RegisterItemClass.

The control raises SerializeTag and DeserializeTag to enable serialization of complex Id and Tag objects without overriding the item classes. The PropertyName event argument specifies which property is being serialized.

Exporting the Diagram

Export Bitmap Images

You can create a bitmap image representing the current diagram using the CreateImage method. The method returns a **Microsoft.Maui.Graphics.IImage** instance. The following example shows how to export a diagram to a Png image file :

C# Copy Code

```
var image = diagram.CreateImage();
image.SaveAsync(stream, ImageFormat.Png);
```

The MindFusion.Diagramming.Export Namespace contains classes for exporting to Dxf, Pdf, Svg, Wmf, Swf and Visio formats.

Export DXF files

Add a reference to the *MindFusion.Diagramming.Export.dll* to your project. Create a DxfExporter instance and call its Export method, specifying a Diagram object and a file path as arguments. Raster images can be used in DXF by referencing external image files. To export the images displayed in diagram elements, enable the ExportImages property. If your application supports the multiple-line text DXF entity - MTEXT, use the ExportTextAsMultiline property. The following example shows how to create a **DxfExporter** instance and export a diagram:

C# Copy Code

```
var dxfExporter = new MindFusion.Diagramming.Export.DxfExporter();
dxfExporter.Export(diagram, exportPath + "diagram.dxf");
```

Export PDF files

Add a reference to the *MindFusion.Diagramming.Export.dll* to your project. Create a PdfExporter instance and call its Export method. Margins specifies the page margins for all pages in the document. If you need to set the page size and orientation, use the PageSize, and PageOrientation properties. The AutoScale property allows scaling the diagram to fit in one or more PDF pages. If invisible items should be displayed in the PDF, enable the InvisibleItems property. The following example shows how to export a diagram to PDF:

C# Copy Code

```
var pdfExporter = new MindFusion.Diagramming.Export.PdfExporter();
pdfExporter.Export(diagram, exportPath + "diagram.pdf");
```

Export SVG files

Add a reference to the *MindFusion.Diagramming.Export.dll* to your project. Create an SvgExporter instance and call its Export method, passing a Diagram object and either a file path or stream as parameters.

C# Copy Code

```
var svgExporter = new MindFusion.Diagramming.Export.SvgExporter();
svgExporter.Export(diagram, exportPath + "diagram.svg");
```

Export Visio files

Add a reference to the *MindFusion.Diagramming.Export.dll* to your project. Create a new instance of the VisioExporter class and call its Export method passing a Diagram object and the full path to the exported Visio .vdx file as parameters. The exporter requires the *VisioExport.vxt* file to be present in the application's directory or the path specified by TemplatePath. This file contains an XML template data used by the component to generate Visio documents.

ShapeNode objects are mapped to their closest Visio counterparts and DiagramLink instances are exported as Visio connectors of a similar style. In addition, most of the attributes of diagram items such as text, text formatting, fill and line colors and styles are preserved.

TableNode objects are exported as groups of Visio shapes or as Entity shapes. If TableNodes should be exported as groups of Visio rectangular shapes, enable the ExportTablesAsGroups property.

C# Copy Code

```
var visioExporter = new VisioExporter();
visioExporter.Export(diagram, exportPath + "diagram.vdx");
```

Export Excel files

The ExcelExporter class from *MindFusion.Diagramming.Export.dll* assembly exports diagrams to Excel Open XML Format (XLSX) files. Diagrams are exported as Excel AutoShapes drawings. The Export overload that takes a DiagramDocument parameter creates a sheet for each DiagramPage in the document. The Export(Diagram) overload creates a single Excel sheet.

Appearance

Styles and Themes

The style and theme system provides a consistent way to customize the appearance of a Diagram and its items. By using styles and themes, it is easier to achieve coherent look and feel across the entire diagram. In addition, changing the look and feel can be achieved by simply replacing the current theme.

Styles

The style is a set of properties. Styles can be applied to diagrams and individual diagram items and this can potentially affect the values of some properties of the target object. More specifically, if the value of a style-affected property in the target object is **null**, then the value is obtained from the associated style. The style does not necessarily need to provide values for all of its properties. If a styled property is **null** and it is not defined in the associated style, the style system searches for a value in the current theme. A complete value precedence list can be found later in this topic.

The styles are represented by the Style class and its various subclasses. Styles can be applied to diagram objects through the Style property of the Diagram class and to diagram items through the Style property of the DiagramItem class. Styles can also participate in themes, in which case they affect all objects of specific type.

Themes

The theme is a set of styles, each associated with specific diagram or item type. When the theme is applied to a diagram, the styles registered in the theme affect all objects of their associated type.

Themes are represented by the Theme class and can be assigned to Diagram objects through the Theme property. Styles are associated with types within the theme through the RegisterStyle method. The following code creates a new theme, with a single style for ShapeNode items and applies this theme to an existing diagram.

C# Copy Code

```
Theme theme = new Theme();
ShapeNodeStyle style = new ShapeNodeStyle();
style.FontFamily = "Courier New";
theme.RegisterStyle(typeof(ShapeNode), style);
diagram.Theme = theme;
```

VB.NET Copy Code

```
Dim theme As New Theme()
Dim style As New ShapeNodeStyle()
style.FontFamily = "Courier New"
theme.RegisterStyle(GetType(ShapeNode), style)
diagram.Theme = theme
```

Alternatively, themes can be loaded from XML files by calling the LoadFromXml method. Such XML files can be produced by calling the SaveToXml method of an existing theme.

Value Precedence

The style and theme system is hierarchical. This means that, when a value of a styled property is requested, the system starts searching for this value along a chain of defined styles. The system returns the first valid value encountered. If no valid value is encountered along the chain, the system returns a globally defined default value for the property. The following list explains the search order in more details:

1. If the local value of the object is not **null**, the local value is returned.
2. If the object has a style and the property in this style has value, this value is returned.
3. If the object is an item, which is part of a diagram, the local style of the diagram is searched.
4. If the object is an item, which is part of a diagram and the diagram has associated theme, the theme is searched for styles registered for the item's type or its base types. If a style with a defined property value is found, this value is returned.
5. Similar search as the previous one is performed for a theme-registered style for the Diagram class.
6. If the above steps do not produce value; the system returns a globally define default for the property.

For example, if the effective value of the text brush property of a ShapeNode is requested (through the EffectiveTextBrush property), the style system performs the following search, according to the above rules:

1. If ShapeNode.TextBrush is not **null**, **ShapeNode.TextBrush** is returned.
2. If ShapeNode.Style is set and the DiagramItemStyle.TextBrush property in this style has value, this value is returned.
3. If the node is part of a diagram, the Diagram.Style is set, and the DiagramStyle.TextBrush property of this style has value, this value is returned.
4. If the node is part of a diagram and the Diagram.Theme is set, the theme is searched for a style registered for the **ShapeNode** type. If there is such style and its **DiagramItemStyle.TextBrush** property has value, this value is returned.
5. Similar search as the previous one is performed for a theme-registered style for the **Diagram** type.
6. Finally, the system returns a globally-defined default value for the **TextBrush** property (which is a solid Black brush).

Brush and Pen Classes

MindFusion.Diagramming defines Brush and Pen classes to represent drawing attributes of shape backgrounds and outlines. Instances of those classes can be assigned to the Brush and Pen properties of each diagram element. There are five specific brush classes that inherit the abstract class **Brush**, namely HatchBrush, LinearGradientBrush, PathGradientBrush, SolidBrush and TextureBrush.

You can set default pen and brush properties like this:

C# Copy Code

```
diagram.ShapeBrush = new MindFusion.Drawing.LinearGradientBrush(
    Color.Yellow, Color.BurlyWood, 20);

MindFusion.Drawing.Brush penBrush = new MindFusion.Drawing.LinearGradientBrush(
    Color.Bisque, Color.Blue, 0);
diagram.ShapePen = new MindFusion.Drawing.Pen(penBrush);
```

After that code runs, every new shape node created will draw with specified pen and gradient background.

Node Effects

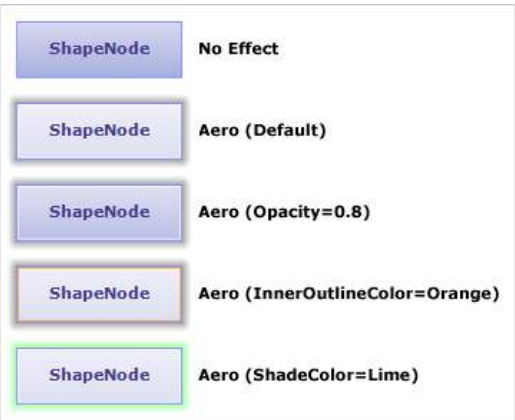
To apply an effect to the nodes in a diagram, create an instance of one of the effect classes and add this instance to the Diagram.NodeEffects collection. To apply the effect to individual nodes, add the instance to the node's Effects collection. To discard the diagram effects for a particular node, add a **null** reference (**Nothing** in Visual Basic) at position 0 in the node's **Effects** collection. Effects can also be applied through styles by adding them to the DiagramStyle.NodeEffects and DiagramNodeStyle.Effects collections respectively.

It is possible to combine different effects by adding instances of their appropriate types to the effect collections. If is also admissible to apply more than one effect of the same type.

The currently supplied effect types are Aero and Glass. They are examined in more details in the sections below.

Aero Effect

The Aero effect is represented by the AeroEffect class. The effect is designed to roughly resemble the Aero theme of Windows. More specifically, the effect adds semi transparency to the nodes, a smooth shade at the outside and a thin inset stroke. The semi transparency is specified through the Opacity property. Opacity varies from 0 to 1 inclusive, where 0 indicates full transparency and 1 indicates the original node color. The semi transparency is applied only on the node brush. The colors of the shade and the inner stroke are specified through the ShadeColor and InnerOutlineColor properties respectively. The following image illustrates how these properties affect the output:



The following example applies the Aero effect on the nodes of an existing diagram:

C# Copy Code

```
AeroEffect effect = new AeroEffect();
effect.Opacity = 0.5f;
effect.InnerOutlineColor = Color.FromArgb(100, Color.White);
diagram.NodeEffects.Add(effect);
```

Visual Basic Copy Code

```
Dim effect As New AeroEffect()
effect.Opacity = 0.5F
effect.InnerOutlineColor = Color.FromArgb(100, Color.White)
diagram.NodeEffects.Add(effect)
```

Tips

- The Aero effect works very well in scenarios with overlapping nodes or in cases where the background is more complex than solid color, due to the effect of semi transparency.
- Aero combines well with the Glass effect. There are some examples at the end of this topic.
- The ShadeColor is usually black or another dark color. However, some nice glow effects can be achieved by setting bright **ShadeColor** on a diagram with dark background.
- The Aero effect can impact performance if there are many nodes visible in the diagram.

Glass Effect

The Glass effect is represented by the GlassEffect class. The Glass effect adds visuals on top of the nodes in order to give the impression of a reflective surface. There are several variations of this effect, which can be selected through the Type property. The color of the reflection and the ambient color can be specified through the ReflectionColor and GlowColor properties respectively. When UsePenAsGlow is set to **true** the color of the node's pen is used as the ambient color. The following image illustrates the differences between the distinct effect types:



The following example applies a Glass effect with the default settings to the nodes of a diagram:

C# Copy Code

```
diagram.NodeEffects.Add(new GlassEffect());
```

Visual Basic Copy Code

```
diagram.NodeEffects.Add(New GlassEffect())
```

Tips

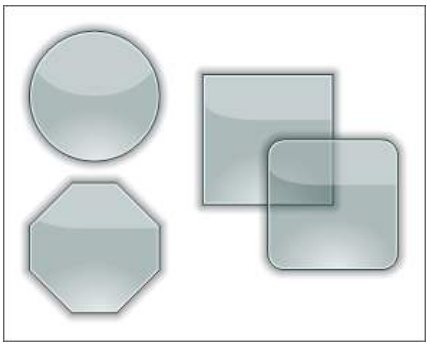
- The GlowColor is usually white. However, for Type4 glass effects it is often more suitable to use black glow.
- The Glass effect combines well with the Aero effect. There are some examples at the end of this topic.
- Several glass effects of different types can be applied simultaneously for emphasized effect.

Examples

The values of the effect properties are usually selected in accordance with the color scheme of the target diagram and the desired end result. In addition different types of effects can be applied simultaneously to achieve more interesting results. The examples below demonstrate possible effect combinations in various color contexts.

Aero & Glass (Dark on Light)

In this example the Aero and Glass effects are combined. The Aero effect is useful in scenarios when dark nodes are placed on light background or vice versa, in order to emphasize the contrast.



Here is the code that applies the above effects:

C# Copy Code

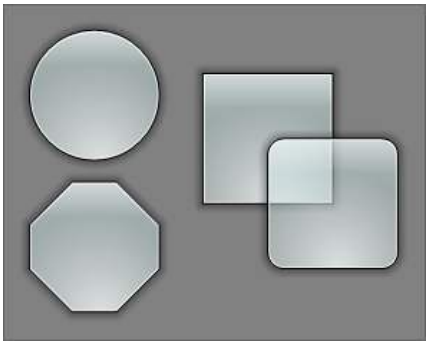
```
diagram.NodeEffects.Add(new GlassEffect());
diagram.NodeEffects.Add(new AeroEffect());
```

Visual Basic Copy Code

```
diagram.NodeEffects.Add(New GlassEffect())
diagram.NodeEffects.Add(New AeroEffect())
```

Aero & Glass (Light on Dark)

This example is similar to the first one, only this time the nodes are painted in light colors and the background is dark.



Here is the code that applies the above effect:

C# Copy Code

```
GlassEffect glassEffect = new GlassEffect();
glassEffect.Type = GlassEffectType.Type3;
diagram.NodeEffects.Add(glassEffect);
diagram.NodeEffects.Add(new AeroEffect());
```

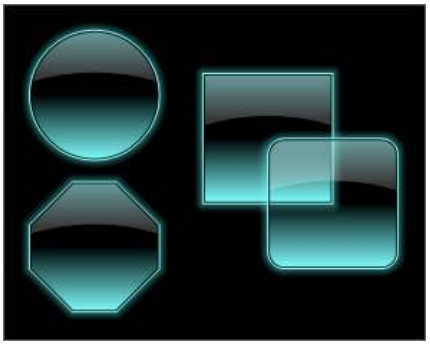
Visual Basic Copy Code

```
Dim glassEffect As New GlassEffect()
glassEffect.Type = GlassEffectType.Type3
```

```
diagram.NodeEffects.Add(glassEffect)
diagram.NodeEffects.Add(New AeroEffect())
```

Glow Effect

This example uses bright colors on black background to create a glow effect. The example utilizes an Aero effect with ShadeColor set to the color of the nodes' pens.



Here is the code that applies the above effects:

C# Copy Code

```
GlassEffect glassEffect = new GlassEffect();
glassEffect.Type = GlassEffectType.Type2;
glassEffect.UsePenAsGlow = true;
diagram.NodeEffects.Add(glassEffect);
AeroEffect aeroEffect = new AeroEffect();
aeroEffect.Opacity = 0;
aeroEffect.InnerOutlineColor = Color.Black;
aeroEffect.ShadeColor = Color.FromArgb(128, 255, 255);
diagram.NodeEffects.Add(aeroEffect);
```

Visual Basic Copy Code

```
Dim glassEffect As New GlassEffect()
glassEffect.Type = GlassEffectType.Type2
glassEffect.UsePenAsGlow = True
diagram.NodeEffects.Add(glassEffect)
Dim aeroEffect As New AeroEffect()
aeroEffect.Opacity = 0
aeroEffect.InnerOutlineColor = Color.Black
aeroEffect.ShadeColor = Color.FromArgb(128, 255, 255)
diagram.NodeEffects.Add(aeroEffect)
```

Programming Interface Overview

The *Scripts/esm* folder from distribution contains ES6 code that lets you import JsDiagram classes from respective modules, which are also available as scoped *@mindfusion/diagramming* package on npm. There are also ES5-compatible scripts located in *Scripts/umd* folder, which are transpiled from ES6 code, and whose classes are exposed as members of global MindFusion namespace object.

The **Reference** section of this help file lists the classes under namespaces (i.e. ES5 compatible hierarchy) whose names you can easily translate to scoped module names when coding for ES6. E.g. if a class is listed under MindFusion.Graphs namespace, you can import it from respective @mindfusion/graphs package.

Diagram Structure

Standard Item Types

Diagram elements in MvcDiagram are represented by DiagramNode and DiagramLink instances. Combined and linked together, they can form complex data structures such as process or entity-relationship diagrams, class hierarchies, networks and graphs. Each of their rich features can be customized through numerous methods and properties.

Shape Nodes

ShapeNode objects can represent geometric shapes, flowchart elements, icons in network diagrams, and so on. They can display both graphics and text in different fonts, sizes and colors. A node's shape is specified by setting its shape property. There are more than 100 predefined node shapes and apart from them, MvcDiagram offers an easy way to define custom shapes. Shape nodes can be rotated, either interactively using the round selection handle drawn above them, or programmatically by setting their rotationAngle property.

SVG Nodes

The SvgNode class represents nodes that can display SVG drawings. SVG graphics are rendered instead of image, and on top of the geometry rendered by the base ShapeNode class. Enable the transparent property to hide the node's geometry and leave only the drawing visible. The SVG file is loaded and associated with the node through an SvgContent object, which should be initialized and then assigned to the content property.

Composite Nodes

Nodes in MindFusion.Diagramming can be composed from components. Components are graphical widgets similar to controls, which can be arranged in hierarchies and added to a Diagram through instances of the CompositeNode class. The component system contains various built-in components, such as Text and Image. Component hierarchies can be defined and loaded through Json templates using loadTemplate and classFromTemplate methods.

Table Nodes

A TableNode displays tabular data, and can be used to represent table schemas in database diagrams or classes in class-hierarchies. A table contains cells arranged in a grid, and every cell can display text and/or image. Cells, columns and rows can be customized in various ways - there are properties available for setting their extents, style, image and text alignment. Tables' dimensions can be programmatically changed; inserting, appending or deleting rows and columns can be done in a very comprehensive, intuitive way whenever you might need it. Tables can also display background images and graphics.

Free-form Nodes

A FreeFormNode collects all points from users' mouse or touch input and displays them as node's outline. To let users draw free-form nodes interactively, set behavior to DrawFreeForms. Use the points property of **FreeFormNode** to get or set outline points programmatically. If the closed property is enabled, the node is drawn as a closed shape and its interior filled, or otherwise the node is drawn as a poly-line. If the distance between first and last points drawn by user is shorter than autoCloseDistance, the node gets closed automatically.

Container Nodes

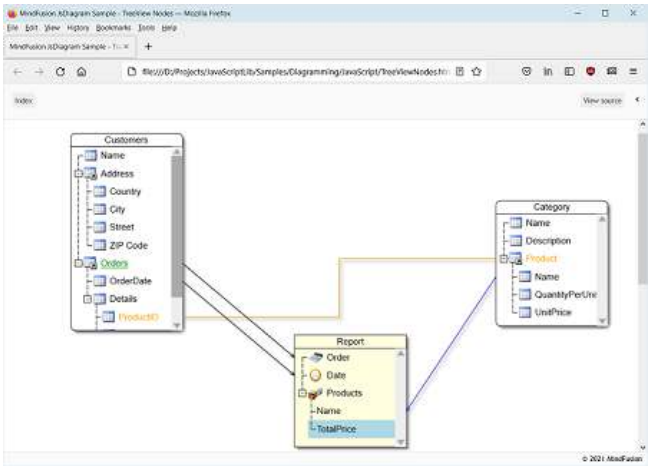
A ContainerNode can contain other diagram nodes. When a container is moved, its child nodes move too. Containers can be nested within each other to unlimited depth. They can be folded to a smaller size, hiding their child items, and unfolded to show content again. Nodes inside a container can be linked to nodes both inside and outside of it. If a link connects items in the same container, the container is not considered an obstacle by the auto-routing algorithm.

Paged Containers

PagedContainerNode is a ContainerNode subclass that organizes its child nodes into a collection of pages, represented by ContainerPage objects. This allows for the creation of more orgainzed diagrams where nodes can be grouped into logical views within a single container. Only the nodes belonging to the currently active page, specified by currentPage property, are rendered and participate in user interactions.

TreeView nodes

The TreeViewNode class represents nodes that can display hierarchical data. The root items displayed in the node can be accessed through the rootItems property. Items can be added and removed individually by using the addItem and removeItem methods, or in bulk by calling the fromObject method, which loads the tree view items from an array of objects.



Video Nodes

VideoNode objects display video streams, along with UI elements to play, pause, seek and change volume. Set node's videoLocation property to specify its video URL. Auto-playing is not supported at this time, and the video can be started only by users clicking the play button. **VideoNode** is built around Video component, which can also be used in CompositeNode templates.

Control Nodes

ControlNode objects display custom HTML content, specified via their template property. A node's DOM structure is created inside the Content div element, which is then rendered on top of the diagram canvas. Note that this prevents **ControlNode**s drawing in diagram' usual Z order, and they will always appear on top of other diagram items. By default the diagram intercepts mouse events of all hosted html elements to enable moving nodes or drawing links. You can specify that interactive elements, such as buttons or text boxes, should receive input instead by setting their **data-interactive** attribute to **true**.

Links

DiagramLink objects are used to represent various types of connections and relations in diagrams. MvcDiagram offers three basic link shapes, which can be assigned to links' shape property:

- polyline links composed of straight-line segments;
- curved links composed of Bézier-spline segments;
- cascading links composed of horizontal and vertical segments, alternating their orientation so that every segment is orthogonal to its neighbor.

A link can be connected to a node at each of the link's ends. It is also possible to have unconnected links - having one or both of their ends not connected to nodes. Some customizable link's properties are: arrowheads, style, text, pen width and style, modification behavior and number of segments.

See Also
Controlling User Interaction

Composite Nodes

The CompositeNode class makes it easier to add custom graphics primitives to nodes. Appearance of composite nodes is defined by a tree of layout panels and child components. Unlike MindFusion's desktop diagram libraries, node templates are specified using JSON notation instead of XML. A sample node class derived from **CompositeNode** can be defined declaratively as below.

JavaScript Copy Code

```
var OrgChartNode = CompositeNode.classFromTemplate("OrgChartNode",
{
 component: "GridPanel",
 rowDefinitions: ["*"],
 columnDefinitions: ["22", "*"],
 children:
 [
  {
   component: "Rect",
   name: "Background",
   pen: "black",
   brush: "white",
   columnSpan: 2
  },
  {
   component: "Image",
   name: "Image",
   autoProperty: true,
   location: "icon4.png",
   margin: "1",
   imageAlign: "Fit"
  },
  {
   component: "StackPanel",
   orientation: "Vertical",
   gridColumn: 1,
   margin: "1",
   verticalAlignment: "Near",
   children:
   [
    {
     component: "Text",
     name: "Title",
```

```
     autoProperty: true,
     text: "title",
     font: "Arial bold"
    },
    {
     component: "Text",
     name: "FullName",
     autoProperty: true,
     text: "full name",
     pen: "blue",
     padding: "1,0,1,0"
    },
    {
     component: "Text",
     name: "Details",
     autoProperty: true,
     text: "details",
     font: "Arial 3"
    }
   ]
  }
 ]
});
```

Objects from the MindFusion.Drawing namespace, such as Rect, Image, Path, Video, double as **CompositeNode** components. You can create custom components by deriving from ComponentBase and assigning a name to the custom class by calling registerComponent. Apart from JSON templates, the visual tree can also be created by adding panels and drawing objects programmatically starting from CompositeNode.root.

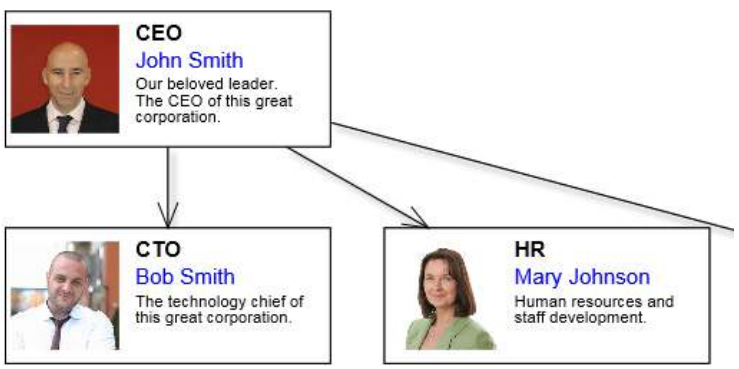
Panel classes include:

- SimplePanel, laying out its children to overlap each other, taking up all available space (useful for background graphics or images).
- StackPanel, laying out its children side by side, either horizontally or vertically.
- GridPanel, laying out its children in rows and columns, where children in same cell overlap. Rows and columns can be either set to a fixed size, auto-size to their child preferred size, or take up all space remaining unoccupied by other rows and columns.

Named components (with specified name value) can be accessed by calling the getComponent method. Components whose autoProperty attribute is enabled will have a pair of get/set property methods generated automatically by classFromTemplate method. The auto-properties from above template can be used as below.

JavaScript Copy Code

```
var node1 = new OrgChartNode(diagram);
node1.setBounds(new Rect(25, 15, 60, 25));
node1.setTitle("CEO");
node1.setFullName("John Smith");
node1.setDetails(
    "Our beloved leader. \r\n" +
    "The CEO of this great corporation.");
node1.setImage("ceo.png");
diagram.addItem(node1);
```



Auto properties are also automatically serialized in JSON format, cloned by node.clone() method and saved/restored by undo commands.

The Shape class used to specify ShapeNode geometry can also be used as a component. When its **isOutline** attribute is set, the shape will control **CompositeNode**'s geometry too, defining hit test and clip area, and link alignment points along node's border. If

isOutline is disabled, the shape will serve mostly as a decorative element. The code below shows a sample fragment specifying shape component in template:

JavaScript Copy Code

```
{
    component: "Shape",
    id: "Cloud",
    autoProperty: true,
    name: "OutlineShape",
    pen: "gray",
    brush: "Transparent",
    isOutline: true
},
```

A Button component that raises event when clicked can be added to template like this:

JavaScript Copy Code

```
{
    component: "Button",
    brush: "#ccc",
    text: "Delete",
    width: 30,
    cornerRadius: 3.5,
    clickHandler: "onDeleteClick"
}
```

Creating and Deleting Items

New items can be added programmatically to the diagram by calling the constructors of the item's class, such as ShapeNode or DiagramLink, and adding the new instance via the addItem method. A shortcut to creating new items is provided by the methods of the Factory class. An instance of that class is returned by the factory property of Diagram. Factory methods such as createShapeNode and createDiagramLink accept respectively node position or link's origin and destination arguments, create an instance of the class and add it to the diagram. The **Factory** class can create instances of JsDiagram predefined item types, while **addItem** can be used to add instances of custom types as well.

JavaScript Copy Code

```
var ShapeNode = MindFusion.Diagramming.ShapeNode;
var Rect = MindFusion.Drawing.Rect;

function initDiagram(diagram)
{
    // create the start node
    var startNode = diagram.factory.createShapeNode(10, 10, 30, 20);
    startNode.text = "start";

    // create the end node
    var endNode = new ShapeNode(diagram);
    endNode.bounds = new Rect(10, 60, 30, 20);
    endNode.text = "end";
    diagram.addItem(endNode);

    // connect the nodes
    diagram.factory.createDiagramLink(startNode, endNode);
}
```

If an existing item should be removed from the diagram, call the removeItem method. The clearAll method clears diagram contents by deleting all items.

By default, deleting a node also removes its links. Set the autoDeleteLinks property to **false** to prevent that, and disconnect the links instead.

Accessing Items

Items in the diagram are exposed via its nodes and links members. **nodes** contains all DiagramNode -derived objects that have been added to the diagram, and **links** contains all DiagramLink objects. By iterating these arrays, you can access all diagram elements and

apply common attributes to them:

JavaScript Copy Code

```
// paint all nodes with light blue color
for (var node of diagram.nodes)
{
    node.brush = "lightBlue";
}

// set the links' arrowheads to triangles
for (var link of diagram.links)
{
    link.headShape = "Triangle";
}
```

Individual items provide methods that let you access their related items in the diagram. **DiagramNode** exposes the links coming into or going out of nodes via the incomingLinks and outgoingLinks properties. The nodes connected by a link can be accessed by means of the origin and destination properties of **DiagramLink**.

The component lets users select multiple diagram items of interest. The selection method of Diagram returns a Selection instance representing currently selected items. The **Selection** class provides nodes and links arrays containing respectively **DiagramNode** and **DiagramLink** -derived objects.

Collapsing and Expanding Tree Branches

If a diagram represents a tree structure, the tree branch starting from a node can be collapsed or expanded again in order to hide or display hierarchy details. To enable this, set the expandable property. To get or set whether a tree branch is expanded or collapsed, use the expanded property of its root node.

Displaying Expand/Collapse Button

If the expandable property of a node is set to true, there is a [±] button displayed beside the node to allow users to collapse or expand the tree branch starting there. By default, pressing that button automatically toggles the expanded property of the node and raises the treeExpanded or treeCollapsed events.

Defining Tree Direction

A node can be connected to other nodes through outgoing and incoming links. Setting the expandOnIncoming property specifies which items connected to a node will be regarded as its descendants. If **expandOnIncoming** is **false** (default value) then each node will be regarded as a parent of its outgoing links and the items attached to them. If **expandOnIncoming** is set to **true**, the node's incoming links define its children. So, this property specifies the direction of parent-children relationship between diagram items and respectively the overall direction in which collapsing / expanding is done.

Attaching and Grouping Items

MvcDiagram allows attaching a node to another node, establishing a subordinate - master relationship between them. The easiest way to attach nodes is to call the attachTo method. When a master node is moved, its subordinated nodes follow it, so that the initial distance between the nodes stays constant. This happens both when a user moves the master node around, and when the node position is changed using a method or a property.

Note

Recent versions of the API include NodeLabel and LinkLabel classes, which address several of the scenarios described below. We recommend using these classes when appropriate, as they offer a more lightweight and easily configurable alternative to attached nodes.

Possible Uses

- Attach a transparent shape node to another node and set its text, so that the shape node acts as an additional label assigned to the master node.
- Attach a shape node to another node and set its image, so it acts as an additional icon displayed inside or near the master node.
- Attach a node to another node and handle nodeClicked or similar events to perform some action on the master node. This makes the subordinated node act as a button or another type of UI element attached to the master node.
- Attach child elements in a tree branch to the branch root, so that when a user moves a tree node, the node's children moves simultaneously with it. Modifying an item in such hierarchical group makes the modification pervade through all dependant subordinated items.
- In all previous scenarios, the locked property can be set for attached nodes which are used only as decorations embellishing the main item. That prevents users from selecting, modifying or drawing links to the subordinate nodes.

Hierarchical Grouping

Every node in a group can be the master node of another group, that is, a node can be subordinate in one group and meanwhile be another group's master. MvcDiagram doesn't impose any limits to the depth and width of group hierarchies built that way. However, it is not possible to put a node as a subordinate in more than one groups because the hierarchy relations would become ambiguous.

Unions

It is possible to create such groups, that if any item in the hierarchy is modified, all other items are updated too. This can be done by putting the items in circular chain of dependencies - that is, item1 is attached to item2, item2 to item3 ... itemN to item1. If any of these items is modified, the change will propagate through the chain until it reaches its originator.

Accessing Group Items

The master node in a group can be accessed using the masterNode property of attached nodes. The array returned by attachedNodes getter on a master node contains references to all subordinate nodes attached to it.

User Interaction

Diagram Events

The diagram raises various events to notify you of the effects of user's actions, such as nodeCreated, nodeModified, nodeDeleted, etc. Events are represented by the EventDispatcher class and are accessible as properties of the corresponding objects (Diagram or DiagramView). Event listeners are attached through the addEventListener method and removed through the removeEventListener method.

The following sample code attaches an event listener to the **nodeCreated** event of a Diagram control. The variable diagram identifies a **Diagram** instance and **onNodeCreated** is the name of the event handler function:

JavaScript Copy Code

```
diagram.nodeCreated.addEventListener(onNodeCreated);
```

All event handlers receive two arguments. The first '*sender*' argument specifies the object that emits the event, and the second '*args*' argument is an instance of an EventArgs -derived class:

JavaScript Copy Code

```
function onNodeCreated(sender, args)
{
    // ...
}
```

While some events are raised to notify you that the user has performed an action, like the **nodeCreated** event, validation events let you accept or reject user actions before actually changing the diagram. Validation event names usually end in '-ing' suffix, such as nodeCreating. The *args* parameter passed to their handler functions is an instance of a CancelEventArgs -derived class. Validation event handlers are expected to set the value of args' cancel property either to **true** if it should be rejected or **false** (the default) if it should be accepted.

The following code handles the nodeDeleting event and, if the node to be deleted is a TableNode instance, cancels the action:

JavaScript Copy Code

```
diagram.nodeDeleting.addEventListener(
    (sender, args) =>
    {
        args.cancel = args.node instanceof TableNode;
    });
```

Controlling User Interaction

There are several ways to control user actions on the diagram as a whole, or on individual items in the diagram.

Drawing Modes

The behavior property of DiagramView provides several predefined sets of rules specifying how to interpret user input through pointing device. The default **behavior** is LinkShapes, where drawing with mouse or touch from an unoccupied point of the document is processed as a ShapeNode creation request, and drawing from a node creates a DiagramLink. Additional members of the **Behavior** enumeration provide modes for drawing different node types, and ones that allow only limited interactions, such as **SelectOnly** or **Modify**.

Item Constraints

Constraints upon how users are allowed to modify a node are set through the constraints and enabledHandles properties. The former can be used to constrain the move direction, node size and aspect ratio. The latter enables or disables specific manipulation handles of a node. That allows making the node non-moveable, non-resizable, or resizable only horizontally/vertically.

Items can be locked by setting their locked property. Locked items are excluded from selection and users cannot select, move or modify them.

In-place Text Editing

The text of nodes and links can be edited in-place by double-clicking the respective items. To enable this feature, set the allowInplaceEdit property to **true**. Depending on the type of item, the control raises either the nodeTextEdited, cellTextEdited or linkTextEdited event when its text has been edited. In-place editing mode is exited when users click outside the text entry control. Pressing ENTER or ESC exits inplace-edit mode if respectively inplaceEditAcceptOnEnter and inplaceEditCancelOnEsc are enabled.

By default, the control creates a **textarea** DOM element over an item to allow editing the item's text. If you prefer using a different type of element as in-place editor, handle the createEditControl event and set the control property of the provided InplaceEditEventArgs object.

Validation Events

While the user draws new nodes or links, the control raises the nodeCreating or linkCreating events after each change of the mouse position. By setting the cancel property of the event object, you can prevent the operation from being completed at the current position but let the user continue drawing. Calling the cancelDrag method immediately stops the operation. The nodeModifying and linkModifying events provide the same functionality during modification of existing objects. Finally, if the user tries to delete an item, you can handle the nodeDeleting or linkDeleting event to prevent this.

Interaction Properties

- allowSelfLoops property of Diagram specifies whether users can draw reflexive / self-loop links.
- allowUnconnectedLinks property specifies whether users can create links without connecting them to nodes (e.g. for simple line or curve graphics in a drawing).
- allowLinksRepeat specifies whether user can draw a second link between same pair of nodes.
- delKeyAction specifies how the DiagramView handles the *Delete* key.
- modifierKeyActions allows changing the function of modifier keys.
- leftButtonActions, middleButtonActions and rightButtonActions allow mapping interaction types to mouse buttons.
- mouseWheelAction allows changing the function of the mouse wheel.
- allowIncomingLinks and allowOutgoingLinks properties of DiagramNode specify if a node accepts respective types of links.
- setting the locked property of a DiagramItem prevents users from interacting with it altogether.

Multi-touch Support

The control handles DOM Pointer events to implement multi-touch interactions, such as zooming, node rotation or simultaneous drawing of multiple diagram items:

- If multiTouchZoom property is enabled (default), the view can be zoomed or panned using two-touch pinch / flick gestures.
- If multiTouchModify property is enabled (default), diagram nodes can be moved, scaled and rotated using two-touch pinch / flick gestures.
- If **multiTouchZoom** property is disabled, each touch draws diagram items corresponding to current behavior.
- If **multiTouchModify** property is disabled, each touch started from a node draws a diagram link.

Latter modes can be used for collaborative whiteboarding / classroom scenarios.

Built-in multi-touch can be disabled altogether by setting enableMultiTouch to **false**, e.g. if you need to handle touch events yourself or through a third-party gesture library. If disabled, the control will revert to its mouse event handling code from previous versions. It will also no longer be able to capture mouse input during drag operations (to detect mouse pointer leaving the view), which is handled through DOM Pointers API.

User-interaction controllers

All kinds of user interaction are carried out by controller objects. Built-in controllers include CreateNodeController, CreateLinkController, ModifyNodeController, ModifyLinkController, PanController. You can also create custom controllers by deriving and overriding the methods of SinglePointerController:

JavaScript Copy Code

```
SinglePointerController.prototype =
{
    start: function (position) {},
    move: function (position) {},
```

```
    validate: function (position) {},
    commit: function (position) {},
    cancel: function (position) {},
    drawInteraction: function (context) {}
}
```

The controller instance used by the diagram when interaction starts is returned by createController method of BehaviorBase -derived classes. You can replace the method of standard behavior classes with a custom function to return a different controller, or inherit a new class from **BehaviorBase** that implements it and assign it to the **currentBehavior** property of DiagramView.

Alignment Grid

MindFusion.Diagramming provides an alignment grid to help users place the diagram items more precisely. To activate the grid, set the alignToGrid property to **true**. If the grid is active while an item is being created or modified, the item's control-points are aligned to the nearest grid points. The distance between adjacent grid points is set by the gridSizeX and gridSizeY properties. The gridOffsetX and gridOffsetY properties allow specifying the horizontal and vertical offset of the first point in the alignment grid.

Items created or modified by a user are automatically aligned to the grid. However, if items are placed in the diagram programmatically, their coordinates are set to the specified positions without aligning. In such cases use the alignPointToGrid method if items must be placed exactly at grid points.

The grid can be either visible or invisible, depending on whether the showGrid property is enabled. If visible, the grid is painted as a matrix of points or as series of crossing lines, as specified by the gridStyle property. gridColor defines the color with which the alignment points or lines are painted.

Undo and Redo

If the undoEnabled property is set to **true**, the control records users actions and allows undoing them later. The records are stored in a list, and can be undone consecutively by calling the undo method. An action that has been undone can be redone by calling the redo method. You can enable or disable UI elements that undo or redo commands by binding them to the **canUndo** and **canRedo** properties.

The control also runs and saves in the undo history AddItemCommand and RemoveItemCommand commands when items are programmatically added or removed from the Items collection of the Diagram. No other changes are recorded automatically when done from code. If support for undo/redo of such changes is required, they must be done through the executeCommand method.

To allow undoing changes of DiagramItem properties, create a ChangeItemCommand instance before setting the property values, and call **executeCommand** afterwards. Any other types of undoable changes can be implemented by deriving from the Command class and overriding its execute, undo and redo methods.

Scrolling and Zooming

The DiagramView control implements scroll and zoom support. If diagram's size is larger than current viewport, scrollbars appear and let the user scroll to clipped parts of the diagram.

Diagram Size

The diagram size is set through the bounds property. Its value is a Rect instance that specifies the origin of the diagram coordinate system and the diagram size. Scrollbar ranges also depend on the current zoomFactor.

Automatic Scrolling

The autoScroll property controls automatic scrolling. If it is enabled and the user drags an adjustment handle near the DiagramView edges while manipulating an item, the control will scroll the viewport towards the mouse position. The scrollZoneSize property defines the size of the area near the view borders that triggers automatic scrolling.

Programmatic Scrolling

In order to scroll the document content programmatically, you can call the scrollTo method or set scrollX and scrollY properties. The latter can also be used to access the current scroll position in case you need it. The bringIntoView method scrolls the view to ensure that a specific diagram item can be seen inside the viewport.

Zooming

Zooming is controlled via the zoomFactor property. The zoom level is expressed as a percentage, 100% meaning no zoom. Higher than 100% makes the diagram items look closer and bigger on the screen - zooming in. Less than 100% makes items look further and smaller on screen - zooming out. To zoom and scroll the view to fit the diagram contents to the control's client area, call zoomToFit. To zoom and scroll the view so it fits a specified rectangular document area to the control's client area, call zoomToRect.

Lasso Zoom Tool

The control supports several ways to zoom using lasso tool:

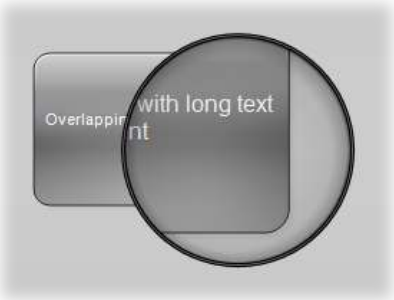
- Set the behavior property to Zoom to always draw zoom lasso.
- Set the Zoom flag for one of the keys in modifierKeyActions to let users draw zoom-lasso with respective modifier key pressed and the left mouse button.
- Assign Zoom value to rightButtonActions or middleButtonActions property to zoom using respective mouse button.

Zoom control

The ZoomControl class lets user change interactively the current zoom level and scroll position of a DiagramView. To set it up, add a <canvas> element to the page (possibly placing it on top of **DiagramView**'s one), call ZoomControl.create to instantiate, and set the control's target property to that view. Set the zoomStep and scrollStep properties to specify the amount added to view's zoom level or scroll position by **ZoomControl**'s buttons. You can customize the control's appearance by setting properties such as fill, borderColor, cornerRadius and tickPosition. For an example, see the *Controls.html* and *Controls.js* sample files from distribution.

Magnifying

The DiagramView provides an option for zooming in (or out) portions of the diagram through the Magnifier tool. The magnifier tool can be permanently activated by setting magnifierEnabled to **true**. Alternatively, on-demand magnification through the right mouse button or by pressing a modifier key can be enabled be setting the appropriate value to the rightButtonActions or modifierKeyActions properties respectively.



The magnification factor can be specified through the magnifierFactor property. The size of the tool and its border thickness can be set through the magnifierWidth, magnifierHeight and magnifierFrameThickness properties.

Panning

The diagram view can be panned by holding ALT while dragging the mouse with the left button down. Another modifier key can be used for panning by setting modifierKeyActions. The middleButtonActions and rightButtonActions properties let you use the middle or right mouse button to start panning. To let users pan the view with the left mouse button without using a modifier key, set the behavior property of DiagramView to Pan or **PanAndModify**.

Testability

User interaction can be tested automatically in several ways outlined below.

- Create a MouseInputDispatcher instance and call its methods

  JavaScript Copy Code

  ```
  onMouseDown: function (position, button)
  onMouseMove: function (position)
  onMouseUp: function (position, button)
  ```

  where position is a Point instance specifying logical coordinates in the diagram (in current measure unit, not scaled by zoom factor). The results of called mouse methods will depend on diagram's current behavior, e.g. you could assert that a sequence of down, move andup events creates a ShapeNode when **behavior** is set to DrawShapes.
- Create an instance of controller class and call its start, move and commit methods. This would not depend on the value of diagram's **behavior**, but could be still affected by other behavioral properties such as allowSelfLoops and allowUnconnectedLinks.
- Check assertions after replaying a recording on a saved diagram, using record, stopRecording and replay methods.

Placement and Layout

Items Placement and Z-order

Measure Unit

The measureUnit property specifies the unit of measure for all location and size-related values assigned to JsDiagram properties or passed as arguments to its methods.

Location and Size of Nodes

Set the bounds method to change the location and size of a node. Its values are Rect objects that contain coordinates and dimension properties. The **bounds** getter lets you find the current location and size of a node as set by users or layout algorithms.

Links Control Points

Links must have two or more control points, depending on their shape property. You can get access to the control points via link's controlPoints collection. You must call updateFromPoints after programmatically moving control points of a link; that's needed in order to repaint the link and update its internal state variables, such as arrowhead angles.

Link Dynamics

The dynamic property of a link can be enabled to make the link automatically recalculate its end point positions if the nodes connected to the link are moved or resized; the end points are moved in such a way that the link points to the centers of the connected nodes. Links also expose an autoRoute property that enables or disables automatic routing for a link. Routed links avoid nodes by finding the shortest path between their origin and destination that does not cross any other nodes.

Interactive Moving or Resizing

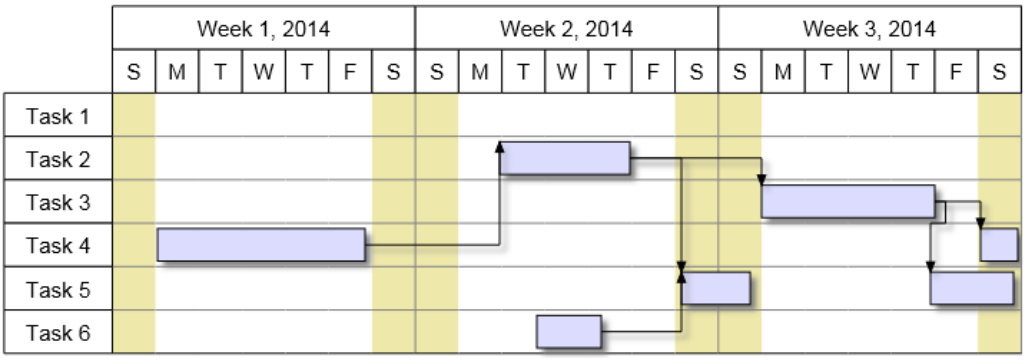
End-users of your application can move or resize the diagram elements. They do this by selecting an item and dragging any of its adjustment handles, a point inside its interior, any of its sides or a selection frame. The handlesStyle property of nodes controls how interaction can be performed. Additionally it defines the representation of selected items on screen. enabledHandles lets you disable specific manipulations users are allowed to perform on a node, for example resizing the item horizontally or moving it. If an item is modified interactively, the nodeModified or linkModified event is raised, depending on the item's type.

Controlling Z-order

The Z-order defines how items are ordered along the Z-axis, so that one item appears in front of or behind another one. Diagram items are sorted by the value of their zIndex property, and rendered on screen in ascending order. You can change the Z index of a DiagramItem by setting **zIndex** (values don't have to be consecutive).

Lane Grid

The classes from the MindFusion.Diagramming.Lanes namespace let you develop process maps and lane diagrams, useful in business process planning and reporting. Processes consist of consecutive operations accomplished by people involved, technical installations, means of contact, and flow of information. The MvcDiagram lane grid features allow indicating and arranging related diagram items in complicated processes and systems, that is, Swim lane diagrams (known as Rummler-Brache diagram), Gantt Charts (planning and scheduling of projects). A lane in the grid can visually contain nodes, links, shapes or images, that represent tasks fulfilled in specified periods of time by people, companies or departments.



The **MindFusion.Diagramming.Lanes** namespace provides the Grid, Header, Cell and Style classes, and the ResizeType enumeration. These types expose additional properties and methods that let you define your own grids as required by your application logic. Grids can include varying number of lanes, divided into child rows and columns whose intersections form cells. The cell color and style can be customized too.

Use the showLaneGrid property of the Diagram class to display lanes in your application. The laneGrid lets you setup the rows and columns in the grid by defining horizontal and vertical lanes, where each lane can be titled by its own header and/or subheadings. The grid can be resized by setting the columnCount and rowCount properties or by adding headers via the addColumnHeader and addRowHeader methods. A column or row can be divided into child elements by adding subheaders, which lets you form a complex hierarchy of unlimited depth. The intersection of rows and columns can be either a simple cell for bottommost elements in the hierarchy, or a group of cells for parent rows and columns. Cells or cell groups can be accessed via the **Grid**'s get method, that lets you index cells either by their integer positions, or by their **Header** objects.

The position of the grid and grid headers within the diagram is defined via topMargin and leftMargin, which specify the grid offset relative to the diagram bounds. hookHeaders enables lane headers to be anchored to the left or right side of the control's visible area. The sizes of headers at different levels in the lane hierarchy can be set in a single operation by setting rowHeadersWidths and columnHeadersHeights properties.

The appearance of the lanes grid can be customized further via the headersOnTop poperty, which specifies whether headers should appear on top of the Z-order, and the topLeftAreaStyle specifying the appearance of the grid's top-left corner.

The **Grid** class provides many miscellaneous properties and methods that can be used to find grid elements, for determining their bounds and for hit-testing: findColumn, findRow, getCellBounds, getCellColumn, getCellFromPoint, getCellRow, getColumn, getRow, getRowIndex, columnHeaderBounds, getColumnIndex, getHeaderBounds, getRowHeaderBounds, getHeaderFromPoint.

Headers in the lanes grid can be customized by using the following members of the **Header** class:

Member name	Description
height	Gets or sets the height of the header and the related row in the grid.
resizeType	Gets or sets the resize type of the header.
rotateTitle	Gets or sets a value indicating whether rotating of the header caption at 90 degree is enabled.
style	Gets or sets the style of the header.
subHeaders	Gets a collection with the subheaders of the header.
title	Gets or sets the header caption.
width	Gets or sets the width of the header and the related column in the grid.

Anchor Points and Patterns

Anchor Points

For each node you can define one or more points to which links should dock when connecting to that node. Such points are represented by instances of the AnchorPoint class. The position of an anchor point relative to node's bounding rectangle is specified via the x and y properties. Their values are expressed as percentage of the node's width and height to accommodate for moving and resizing.

You can control whether an anchor point accepts incoming or outgoing links or both. That is done via the allowIncoming and allowOutgoing properties. When a link is drawn, its ends are aligned to the nearest anchors allowing docking of the respective end type. Another way to control whether links can be connected to an anchor point is to handle the validateAnchorPoint event.

Anchor Patterns

An AnchorPattern instance defines a set of anchor points to be used together. When creating a pattern, pass to its constructor an array of AnchorPoint objects. The **AnchorPattern** class exposes several predefined patterns as static properties. A pattern can be assigned as an integral entity to the anchorPattern property of nodes. Setting the anchorPattern property of a Shape object lets you define the connection points for all ShapeNodes of that shape. The following example creates a pattern of four anchor points:

JavaScript Copy Code

```
var AnchorPattern = MindFusion.Diagramming.AnchorPattern;
var AnchorPoint = MindFusion.Diagramming.AnchorPoint;

var apat1 = new AnchorPattern([
    new AnchorPoint(50, 0, true, true),
    new AnchorPoint(100, 50, true, true),
    new AnchorPoint(50, 100, true, true),
    new AnchorPoint(0, 50, true, true)
    ]);
```

The points are located at the middles of bounding rectangle sides of nodes to which the pattern is assigned. All four points accept both incoming and outgoing links.

Link Routing

Routed links avoid nodes marked as obstacles by finding the shortest path between their origin and destination that does not cross any other nodes.

Setting Up a Link Router

The control finds paths for links affected by modifications of nearby nodes by calling methods of an object that implements the Router interface. This object can be get or set through the linkRouter property of the Diagram class. JsDiagram provides several classes that implement **Router** - PatternRouter, GridRouter and CompositeRouter. By default **linkRouter** is set to a **CompositeRouter** instance.

Automatic or Explicit Routing

The routeAllLinks method can be invoked to explicitly rearrange links in a way they do not cross nodes. In addition, DiagramLink defines an autoRoute property that enables or disables automatic routing of a link, applied when user moves link's adjacent nodes or draws a new node in its vicinity. **autoRoute** is initialized with the value of diagram's routeLinks property.

Tuning the Routing Algorithm

Router classes define properties that let you fine-tune their specific algorithms. For a reference, see specific CompositeRouter, PatternRouter or GridRouter.

Performance Consideration

The default router is designed to produce visually optimal link paths that avoid overlapping with nodes and minimize bends. However, it is computationally intensive and can impact performance in diagrams with a large number of links or frequent layout changes. For improved performance, you could set linkRouter to a simple Router instance. This will reduce CPU usage and improve responsiveness, though the resulting link paths may be less visually refined.

Laying out the Diagram

The diagram can be arranged automatically by calling its arrange method and specifying a layout algorithm. Different layout algorithms are implemented by the various Layout classes from the MindFusion.Graphs namespace. To apply one of them, create an instance of the respective class, customize its properties and pass it as an argument to **arrange**. It is also possible to lay out the contents of an individual ContainerNode by calling the container's arrange method.

Layered Graph Layout

The LayeredLayout algorithm arranges diagram nodes in layers according to several criteria, most important of which are: connected nodes must be placed close together; links must flow in one direction if possible; links must cross as few layers as possible; links must not cross other links. To apply the layout to a diagram, create a **LayeredLayout** instance, set its members and invoke the arrange method.

Spring-Embedder Graph Layout

Spring-Embedder is force-directed layout algorithm. It simulates a physical system in which nodes repulse each other, and the links between them act as confining springs. Graphs processed by Spring-Embedder have their nodes distributed uniformly across the diagram area. The algorithm can be applied on a graph by creating a SpringLayout object and calling the arrange method. Members of **SpringLayout** allow you to set the desired distance between graph nodes and the number of iteration steps to run while processing the simulation.

Tree Layout

If application's data is structured hierarchically, it is appropriate to apply tree layout on it. To do this, create a TreeLayout or BorderedTreeLayout instance and call the arrange method. The difference between two classes is where they place child nodes in relation to their parent; the former centers the parent above its children, the latter arranges children in columns at a horizontal offset form their parent (see **Tutorial 2**). Members of both classes control the tree direction, how much space to leave between tree levels and between nodes on the same level.

Fractal Layout

The FractalLayout tree layout algorithm places child nodes symmetrically around their parent node. Nodes at the lowest level are arranged directly in a circle around their parent. At the upper level, the already arranged nodes form branches that are arranged in a circle around the new parent node. The algorithm is recursively repeated till the highest level is reached.

Treemap Layout

Tree maps represent hierarchies by nesting child nodes within their parents, where the areas of leaf nodes are proportional to their weight values. Unlike other layout algorithms, TreeMapLayout expects hierarchies to be defined via grouping or containment (see attachTo method and ContainerNode class), and will ignore any links in the diagram.

Flowchart Layout

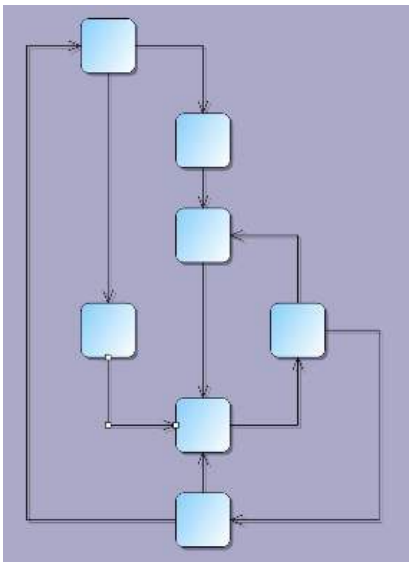
FlowchartLayout recognizes program code -like patterns in the graph, such as loops, sequences and if/switch branchings, and arranges them recursively. **FlowchartLayout** could be used to arrange other types of graphs as well, though with some restrictions. For example it treats all back links as loops in the code, and expects that they are nested - loop links starting closer to the stop node should end closer to the start node. Another similar restriction is that there shouldn't be any cross-links that connect different branches of a decision sub-graph.

Topological Layout

TopologicalLayout applies graph topological ordering to the diagram (topological ordering of a graph is one where link's origin node is placed before link's destination node). The layout algorithm arranges nodes in a row or a column, depending on the value of the direction property, in such a way that there are no backward links when the graph is acyclic. If the graph contains cycles, the algorithms selects ordering with as few backward links as possible. Links that connect non-adjacent nodes are rendered as arcs, whose amplitude is proportional to the distance between respective connected nodes. Forward links are rendered at one side of the nodes and back links are rendered at the opposite side. All that makes it easy to discern graph features such as overall flow direction, cycles and nested cycles.

Orthogonal Layout

The OrthogonalLayout class implements an orthogonal graph layout algorithm. Each link is drawn as a chain of alternating horizontal and vertical segments. Nodes are placed in a way that facilitates few links bends and crossings. This algorithm was designed for planar graphs where the nodes have at most four incident links, and produces best results with such graphs as input. It can arrange any arbitrary graph as well, by adding some dummy nodes and links to planarize the graph and reduce its vertex degree, and removing them after the layout.



Radial Tree Layout

The RadialTreeLayout class arranges tree levels in concentric circles around the root. The levelDistance property specifies the distance between adjacent circles. The direction property specifies placement of first child node. Set stretchFactor to a value larger than 1 in order to stretch level circles into ellipses, providing more space for the layout when using wider nodes.

Grid Layout

The GridLayout algorithm arranges diagram nodes in a grid, keeping connected nodes close to each other. The algorithm strives to achieve a small number of link crossings. It is based on an iterative process whose initial steps shuffle the grid nodes randomly. That can lead to very different results each time the algorithm is run.

Appearance

Customizing Node Appearance

Nodes can have different shape, background image or color. They can also have differently aligned text.

Geometric Shapes

ShapeNode exposes a shape property that specifies the geometric shape of the node. Its value can be set to either a Shape object (returned by one of the static methods of the Shapes class), or a string id of the shape, which coincides with the name of the method that gets it. Here is an example:

JavaScript Copy Code

```
myNode.shape = "Arrow1";
```

TableNode and ContainerNode provide shape properties too, but their shapes can only be set to either rectangle or rounded rectangle, as specified via SimpleShape enumeration.

Bitmap Images

Each node can display a bitmap image as its content. The image can be set either via the image or imageContent properties, where the former accepts DOM Image instances, and the latter accepts base64-encoded bitmaps. By default the bitmap is drawn centered inside the node. You can change the alignment by setting imageAlign. Images can also be loaded by URL via the imageLocation property.

Fill Color

Nodes can be painted in any color that you choose. Set the brush property to specify how a node is painted.

JavaScript Copy Code

```
myNode.brush = "Red";
```

Linear Gradients

Nodes can be filled using linear gradients by assigning objects in following format to brush. If the gradient should blend only two colors, they can be specified using **color1** and **color2** fields instead of **colorStops**.

JavaScript Copy Code

```
var brush1 =
{
 type: "LinearGradientBrush",
 colorStops:
 [
  { position: 0, color: "red" },
  { position: 0.5, color: "green" },
  { position: 1, color: "blue" }
 ],
 angle: 30
};
myNode.brush = brush1;
```

Radial Gradients

Nodes can be filled using radial gradients by assigning objects in following format to brush. If the gradient should blend only two colors, they can be specified using **color1** and **color2** fields instead of **colorStops**.

JavaScript Copy Code

```
var brush2 =
{
 type: "RadialGradientBrush",
 colorStops:
 [
  { position: 0, color: "white" },
  { position: 0.5, color: "lightGray" },
  { position: 1, color: "blue" }
 ],
 x1: 20, y1: 20, radius1: 30,
 x2: 40, y2: 20, radius2: 80
};
myNode.brush = brush2;
```

Text and Font

Nodes can display text with customizable font and horizontal/vertical alignment. Set the text property to specify the text label for a node, and textColor to change its color. textAlignment and lineAlignment properties let you set respectively horizontal and vertical alignment. The font property lets you specify a distinct font for each node:

JavaScript Copy Code

```
var Font = MindFusion.Drawing.Font;

//the bool arguments specify whether the font is bold and italic
startNode.font = new Font("sans-serif", 5, true, false);
```

Apart from main text content, the NodeLabel class lets you display additional labels and pin them to node's edges or corners. Call the addLabel method to create such supplementary labels, and use **NodeLabel** (and base ItemLabel) properties and methods to customize their appearance and position.

The text of ShapeNodes can be laid out and rendered inside a polygonal outline; that depends on the value of polygonalTextLayout.

Styled Text

The enableStyledText property of shape nodes and tables enables or disables styled text rendering. Styled text can have various attributes applied to the characters it contains. The attributes are specified using HTML-like formatting tags embedded in the raw text. The following table lists the currently supported style tags:

Tag	Description
text	Indicates italic text.
text	Indicates bold text.
<u>text</u>	Indicates underlined text.
_{text}	Indicates subscript.
^{text}	Indicates superscript.
<color=#RRGGBB>text</color>	Changes the color of the text. The red, green and blue color channels are specified as two-digit hexadecimal numbers.
 	Indicates line break.
<a=url>text	Renders specified text as a hyperlink and raises hyperlinkClicked event when link has been clicked.
<size=N>text</size>	Changes the font size.

Node Effects

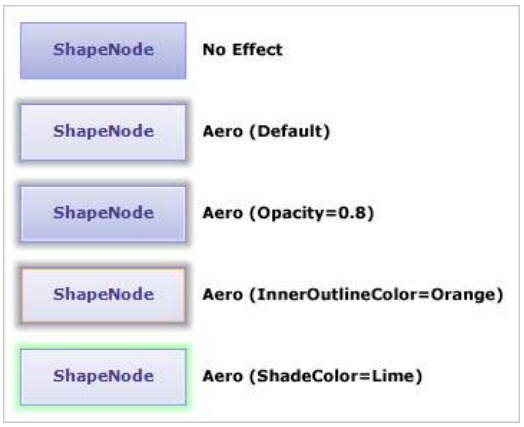
To apply an effect to the nodes in a diagram, create an instance of one of the effect classes and add this instance to the Diagram.nodeEffects array. To apply the effect to individual nodes, add the instance to a node's own effects array. To ignore the global diagram effects for a particular node, add a **null** reference at position 0 in the node's **effects**. Effects can also be applied through styles by adding them to the Style.nodeEffects array of the diagram or a node.

It is possible to combine different effects by adding instances of their appropriate types to the effect collections. If is also admissible to apply more than one effect of the same type.

The currently supplied effect types are Aero and Glass. They are examined in more details in the sections below.

Aero Effect

The Aero effect is represented by the AeroEffect class. The effect is designed to roughly resemble the Aero theme of Windows. More specifically, the effect adds semi transparency to the nodes, a smooth shade at the outside and a thin inset stroke. The semi transparency is specified through the opacity property. Opacity varies from 0 to 1 inclusive, where 0 indicates full transparency and 1 indicates the original node color. The semi transparency is applied only on the node's brush. The colors of the shade and the inner stroke are specified through the shadeColor and innerOutlineColor properties respectively. The following image illustrates how these properties affect the output:



The following example applies the Aero effect on the nodes of an existing diagram:

JavaScript Copy Code

```
var AeroEffect = MindFusion.Diagramming.AeroEffect;

var effect = new AeroEffect();
effect.opacity = 0.5;
effect.innerOutlineColor = "rgba(255,255,255,0.5)";
diagram.nodeEffects.push(effect);
```

Tips

- The Aero effect works very well in scenarios with overlapping nodes or in cases where the background is more complex than solid color, due to the effect of semi transparency.
- Aero combines well with the Glass effect. There are some examples at the end of this topic.
- The shadeColor is usually black or another dark color. However, some nice glow effects can be achieved by setting bright **shadeColor** on a diagram with dark background.
- The Aero effect can impact performance if there are many nodes visible in the diagram.

Glass Effect

The Glass effect is represented by the GlassEffect class. The Glass effect adds visuals on top of the nodes in order to give the impression of a reflective surface. There are several variations of this effect, which can be selected through the type property. The color of the reflection and the ambient color can be specified through the reflectionColor and glowColor properties respectively. When usePenAsGlow is set to **true**, the color of the node's pen is used as the ambient color. The following image illustrates the differences between the distinct effect types:



The following example applies a Glass effect with the default settings to the nodes of a diagram:

JavaScript Copy Code

```
var GlassEffect = MindFusion.Diagramming.GlassEffect;
diagram.nodeEffects.push(new GlassEffect());
```

Tips

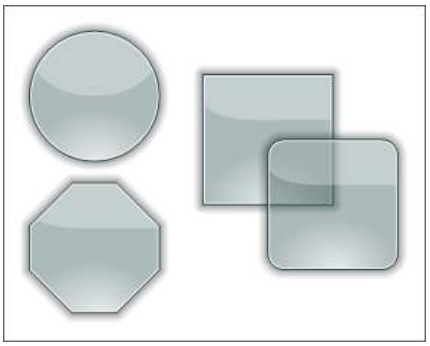
- The glowColor is usually white. However, for Type4 glass effects it is often more suitable to use black glow.
- The Glass effect combines well with the Aero effect. There are some examples at the end of this topic.
- Several glass effects of different types can be applied simultaneously for emphasized effect.

Examples

The values of the effect properties are usually selected in accordance with the color scheme of the target diagram and the desired end result. In addition different types of effects can be applied simultaneously to achieve more interesting results. The examples below demonstrate possible effect combinations in various color contexts.

Aero & Glass (Dark on Light)

In this example the Aero and Glass effects are combined. The Aero effect is useful in scenarios when dark nodes are placed on light background or vice versa, in order to emphasize the contrast.



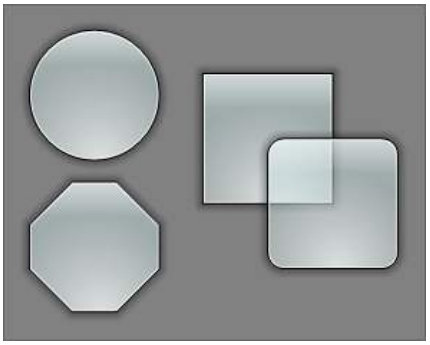
Here is the code that applies the above effects:

JavaScript Copy Code

```
diagram.nodeEffects.push(new GlassEffect());
diagram.nodeEffects.push(new AeroEffect());
```

Aero & Glass (Light on Dark)

This example is similar to the first one, only this time the nodes are painted in light colors and the background is dark.



Here is the code that applies the above effect:

JavaScript Copy Code

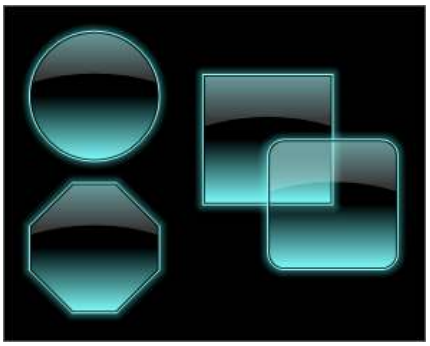
```
var AeroEffect = MindFusion.Diagramming.AeroEffect;
var GlassEffect = MindFusion.Diagramming.GlassEffect;
var GlassEffectType = MindFusion.Diagramming.GlassEffectType;

var glassEffect = new GlassEffect();
glassEffect.type = GlassEffectType.Type3;

diagram.nodeEffects.push(glassEffect);
diagram.nodeEffects.push(new AeroEffect());
```

Glow Effect

This example uses bright colors on black background to create a glow effect. The example utilizes an Aero effect with shadeColor set to the color of the nodes' pens.



Here is the code that applies the above effects:

JavaScript Copy Code

```
var AeroEffect = MindFusion.Diagramming.AeroEffect;
var GlassEffect = MindFusion.Diagramming.GlassEffect;
var GlassEffectType = MindFusion.Diagramming.GlassEffectType;

var glassEffect = new GlassEffect();
glassEffect.type = GlassEffectType.Type2;
glassEffect.usePenAsGlow = true;
diagram.nodeEffects.push(glassEffect);

var aeroEffect = new AeroEffect();
aeroEffect.opacity = 0;
aeroEffect.innerOutlineColor = "black";
aeroEffect.shadeColor = "rgb(128, 255, 255)";
diagram.nodeEffects.push(aeroEffect);
```

Customizing Link Appearance

DiagramLinks-s consist of series of straight or Bezier segments and arrowhead shapes. The link and arrowhead shapes, and also color attributes and text can be customized as detailed below:

Link Geometry

Links can be straight or curved and can contain one or more segments. Set DiagramLink.shape to change the shape. It takes as an argument one of the LinkShape enumeration values:

JavaScript Copy Code

```
myLink.shape = LinkShape.Cascading;
```

The link can be customized further by changing its controlPoints array, whose elements specify the position of the link segments. The number of control points per segment depends on the link's **shape**: **Bezier** links have four control points per segment. Each straight segment has two control points. Adjacent segments share a control point.

The following code sets three control points of a polyline link, which results in two segments:

JavaScript Copy Code

```
var Point = MindFusion.Drawing.Point;
myLink.controlPoints = [new Point(10, 10), new Point(30, 30), new Point(30, 50)];
```

Note

Always call DiagramLink.updateFromPoints() after you change coordinates of individual control points in the array.

Rounded Links

Links comprising straight segments (ones whose shape is set to Polyline or **Cascading**) can draw arcs as joints between their adjacent segments. To enable that feature, set roundedLinks to **true**. Use roundedLinksRadius to specify the radius of the joining arcs.

Link Crossings

Intersection points where links cross their paths can be rendered as arcs or cuts, as specified via linkCrossings. It depends on the Z-order, which links jump over the others, or which links appear cut by the others. If **linkCrossings** is set to Arcs, links with higher

ZIndex jump over those with lower ZIndex. If set to **Cut**, links at lower Z position are cut by links at higher Z. Radius of arcs and breaks can be set through the crossingRadius property.

Customizing Arrowheads

Each link can display arrowhead shapes at its ends. The shape at the start is specified via baseShape property, the shape at the end is specified via headShape. Arrowheads can also be displayed at middles of segments, helping users follow the path of long links. They are defined by the intermediateShape property. Arrowhead values can be set to either a Shape instance or a string that identifies it:

JavaScript Copy Code

```
myLink.headShape = "BowArrow";
```

Shape objects designed specifically as arrowheads are exposed as static members of the ArrowHeads class.

Specifying Colors

Link segments are rendered according to stroke, strokeThickness and strokeDashStyle attributes. The arrowheads at the head and base are filled using headBrush and baseBrush properties:

JavaScript Copy Code

```
// paints the link head with red brush
myLink.headBrush = "Red";
```

Setting Text

Set the text property to specify text label of a link, and font to customize its font. Here is an example how to do this:

JavaScript Copy Code

```
myLink.text = "myLabel";

var Font = MindFusion.Drawing.Font;

// the bool arguments specify whether the font is bold and italic
myLink.font = new Font("sans-serif", 5, true, false);
```

The font size is specified in the diagram's current measure unit, which is millimeter by default.

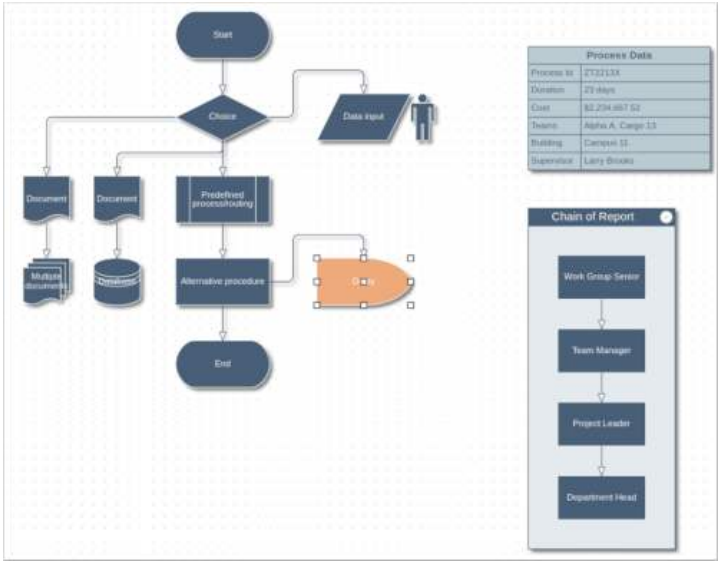
Apart from main text content, the LinkLabel class lets you display additional labels and pin them to link's points or segments. Call the addLabel method to create such supplementary labels, and use **LinkLabel** (and base ItemLabel) properties and methods to customize their appearance and position.

Styles and Themes

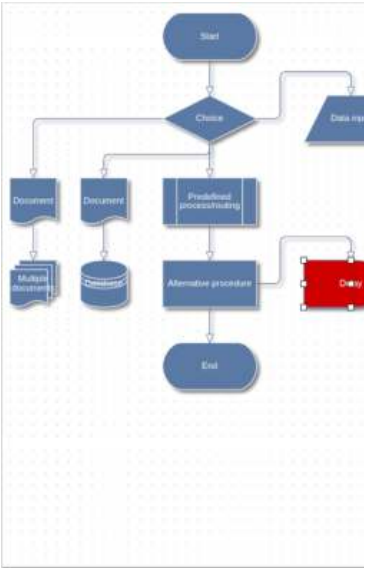
Appearance of items can be customized using Style objects assigned to respective style property of DiagramItem. The attributes from a style are applied only when item's own corresponding properties do not contain values.

Style objects can also be collected into a Theme and applied as a set to the whole diagram by assigning it to the theme property. The library includes nine stock themes listed below, accessible via respective static methods of the **Theme** class.

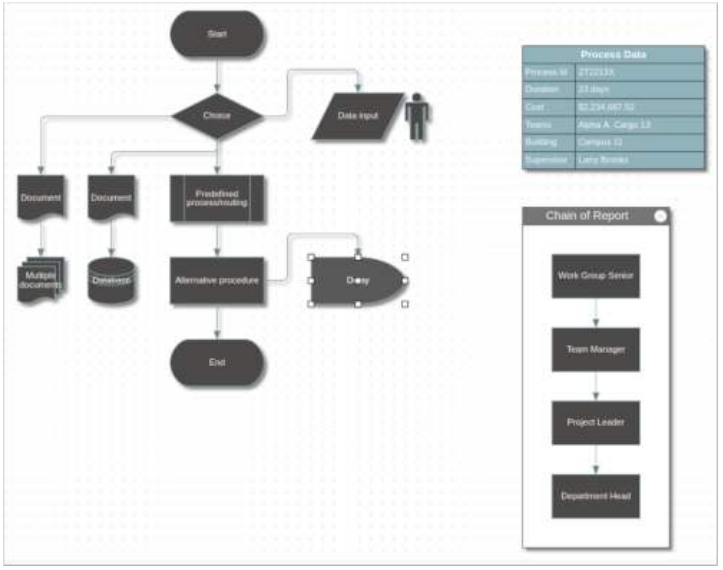
Blue	Business



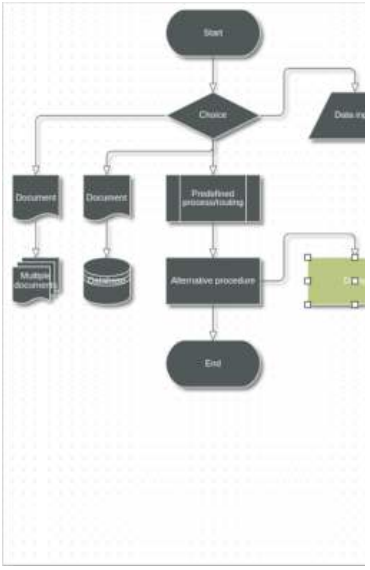
Gray



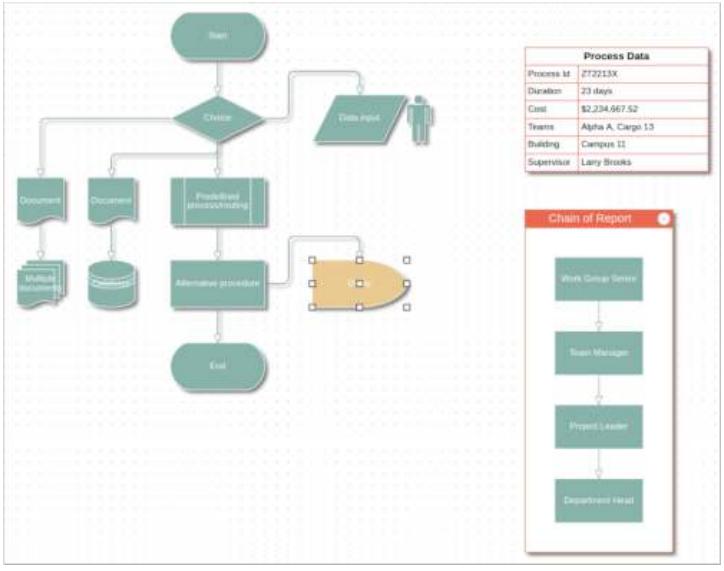
Green

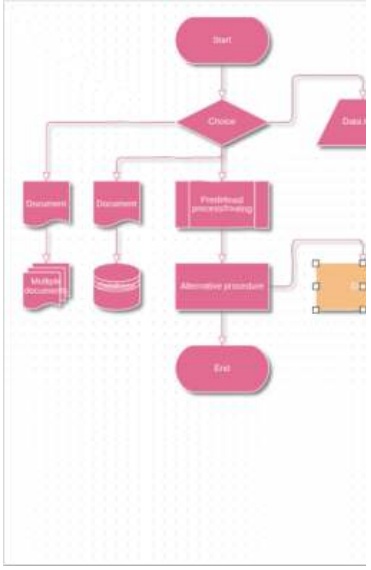


Pastel



Peach





Creating and applying a custom theme looks like this:

JavaScript Copy Code

```
var theme = new Theme();

var shapeNodeStyle = new Style();
shapeNodeStyle.brush = { type: 'LinearGradientBrush', color1: '#e0e9e9', color2: '#9caac6', angle:
90 };
shapeNodeStyle.stroke = "#7F7F7F";
shapeNodeStyle.textColor = "#585A5C";
shapeNodeStyle.fontName = "Verdana";
shapeNodeStyle.fontSize = 3;
theme.styles.set("std:ShapeNode", shapeNodeStyle);

var linkStyle = new Style();
linkStyle.stroke = "#7F7F7F";
linkStyle.textColor = "#585A5C";
linkStyle.fontName = "Verdana";
linkStyle.fontSize = 3;
theme.styles.set("std:DiagramLink", linkStyle);

var tableStyle = new Style();
tableStyle.brush = { type: 'LinearGradientBrush', color1: '#FCFCFC', color2: '#9DABB4', angle: 90
};
tableStyle.stroke = "#7F7F7F";
tableStyle.textColor = "#585A5C";
tableStyle.fontName = "Verdana";
tableStyle.fontSize = 3;
theme.styles.set("std:TableNode", tableStyle);

diagram.theme = theme;
```

CSS Styling

The styling system also lets you define item appearance through external CSS files. You can style different item types using custom element selectors such as **mfd-shapenode**, **mfd-diagramlink**, or **mfd-tablenode**. The system maps CSS Custom Properties to the internal Style properties. For example, use **--brush** for backgrounds, **--stroke** for outlines, and **--font-size** for text. The new cssClass property on DiagramItem allows for more granular styling using standard class selectors in your CSS. To ensure high performance, computed styles are cached globally, and the system automatically resolves style inheritance across class hierarchies (e.g., a ShapeNode will resolve styles from **mfd-shapenode**, **mfd-diagramnode**, and **mfd-diagramitem** selectors in that order).

CSS Copy Code

```
/* Style all shape nodes */
mfd-shapenode {
    --brush: #f9f9f9;
    --stroke: #333;
    --stroke-thickness: 1px;
    --font-name: 'Verdana';
}

/* Specific styling via the new cssClass property */
mfd-shapenode.urgent {
    --brush: #ffcccc;
    --stroke: #ff0000;
    --stroke-thickness: 2px;
}

/* Style table captions using custom properties */
mfd-tablenode {
    --caption-brush: #e0e0e0;
    --text-color: #000066;
}
```

Here is the complete list of **mfd-** prefixed CSS element selectors available for styling items:

Base Elements

- **mfd-diagramitem**: The base selector for all diagram items. Styles applied here will be inherited by all other items unless overridden.
- **mfd-diagramnode**: The base selector for all node types. Inherits from **mfd-diagramitem**.

Specific Node Types

(all inherit from **mfd-diagramnode**)

- **mfd-shapenode**: Styles ShapeNode instances.
- **mfd-svgnode**: Styles SvgNode instances.
- **mfd-compositenode**: Styles CompositeNode instances.
- **mfd-controlnode**: Styles ControlNode instances.
- **mfd-videonode**: Styles VideoNode instances.
- **mfd-freeformnode**: Styles FreeFormNode instances.

Container & Complex Nodes

- **mfd-containernode**: Styles ContainerNode instances (inherits from **mfd-captionednode**).
- **mfd-captionednode**: The base selector for nodes with captions like Tables and TreeViews. Inherits from **mfd-diagramnode**.
- **mfd-tablenode**: Styles TableNode instances (inherits from **mfd-captionednode**).
- **mfd-treeviewnode**: Styles TreeViewNode instances (inherits from **mfd-captionednode**).

Links

- **mfd-diagramlink**: Styles DiagramLink instances. Inherits from **mfd-diagramitem**.

Sample Inheritance Chains

- **mfd-shapenode** ? **mfd-diagramnode** ? **mfd-diagramitem**
- **mfd-tablenode** ? **mfd-captionednode** ? **mfd-diagramnode** ? **mfd-diagramitem**
- **mfd-diagramlink** ? **mfd-diagramitem**

Properties

Here is the list of custom CSS properties (variables) available for diagram items. These properties should be defined within the **mfd-** element selectors, and possibly assigned to specific items via the cssClass property.

Primary Appearance

- **--brush**: Specifies the background / fill color. Supports CSS color strings (e.g., `#FF0000`, `rgba(...)`, `lightblue`).
- **--stroke**: Specifies the outline / border color.
- **--stroke-thickness**: Specifies the width of the outline. Supports units like px, mm, pt, or in (values are converted to diagram's measureUnit).
- **--shadow-color**: Specifies the color of the item's shadow.

Typography & Text

- **--text-color**: Specifies the color of the text label.
- **--font-name**: Specifies the font family (e.g., 'Verdana', 'Arial').
- **--font-size**: Specifies the size of the font. Supports CSS units (values are converted to diagram's measureUnit).
- **--font-weight**: Use 'bold' to apply bold styling.
- **--font-style**: Use 'italic' to apply italic styling.
- **--text-decoration**: Use 'underline' to apply underline styling.

Selection States

- **--selected-brush**: Specifies the fill brush used when the item is selected.
- **--selected-stroke**: Specifies the outline color used when the item is selected.

Specialized Properties

- **--caption-brush**: Specifies the background color of captions (used by **mfd-tablenode**, **mfd-containernode**, and **mfd-treeviewnode**).
- **--vertical-border-stroke**: Specifies the color of vertical grid lines/borders (primarily for TableNode).
- **--horizontal-border-stroke**: Specifies the color of horizontal grid lines/borders (primarily for **TableNode**).

Define Shape Geometries

Diagrams can display geometric shapes by setting the shape property of ShapeNode instances.

Shape Templates

The Shape class provides the means for defining complex node shapes, made of straight lines, arcs and Bézier curves. A shape template must always contain an outline to be used for hit testing, clipping and finding intersections with other items. Optionally shapes can contain decoration elements, which are visual elements that do not take part in hit testing and clipping. JsDiagram provides a set of over a 100 **predefined shape templates**, which can be accessed by the Shapes collection or fromId method of the **Shape** class. A shape is assigned to a node by means of the shape property.

JavaScript Copy Code

```
shapeNode.shape = MindFusion.Diagramming.Shapes.Actor();
shapeNode.shape = MindFusion.Diagramming.Shape.fromId("Actor");
```

Custom Shapes

You can define your own shapes by using the static createShape method of the Diagramming.Controls.ShapeBuilder class. The shape geometry is specified as an array of ShapeElements. The ShapeElement class constructor expects a command and an array of command parameters, and optionally styling. The list of supported commands matches the HTML Canvas drawing API (e.g moveTo, lineTo, bezierCurveTo, arc, roundRect). Coordinates are defined as percentage of a node's bounding rectangle.

JavaScript Copy Code

```
var outline = [];
outline.push(new ShapeElement("moveTo", [0, 33]));
outline.push(new ShapeElement("lineTo", [33, 0]));
outline.push(new ShapeElement("lineTo", [66, 0]));
outline.push(new ShapeElement("lineTo", [100, 33]));
outline.push(new ShapeElement("lineTo", [100, 100]));
outline.push(new ShapeElement("bezierCurveTo", [66, 112, 33, 87, 0, 100]));
outline.push(new ShapeElement("lineTo", [0, 33]));
var shape = MindFusion.Diagramming.Controls.ShapeBuilder.createShape("MyShape", outline);
shapeNode.shape = shape;
```

The Shape definition can also contain decorations and shape decorations. A decoration is a geometry that consists of one or more lines or curves. A shape decoration is a closed path of lines and/or curves, that can also have a fill. A shape can have one decoration and a number of shape decorations, that are specified through the decorationElements and shapeDecorationElements parameters as follows:

JavaScript Copy Code

```
var decoration = [];
decoration.push(new ShapeElement("moveTo", [0, 33]));
decoration.push(new ShapeElement("lineTo", [33, 0], "red"));
decoration.push(new ShapeElement("lineTo", [66, 33], "blue", 4));

var shapeDecorationElements = [];
shapeDecorationElements.push(new ShapeElement("roundRect", [50, 50, 80, 80, 2]));
var shapeDecoration = new ShapeDecoration(shapeDecorationsElements, "red");

var shape = MindFusion.Diagramming.Controls.ShapeBuilder.createShape(
    "MyShape", outline, decoration, [shapeDecoration]);
```

If you don't need the per-element styling options, another way to create a shape is by creating an instance of the **Shape** class providing the outline definition using SVG Path syntax.

JavaScript Copy Code

```
var shape = new Shape({
    id: "MyShape"
    outline: "M100,50 A50,50,0,1,1,0,50 A50,50,0,1,1,100,50",
    decoration: "M98,50 A48,48,0,1,1,2,50 A48,48,0,1,1,98,50"});
```

A string identifier must be specified for every shape so they will be accessible via the Shape.fromId method.

Shape Libraries

You can use a Shape Library xml file to store your collection of custom shapes. Set the shapeLibraryLocation property to specify the path to the file.

JavaScript Copy Code

```
diagram.shapeLibraryLocation = "MyShapes.sl";
```

The shapes in the Shape Library file will be automatically added to the shared list of shapes kept in the Shape class, which lets you access them by calling the fromId method.

Shape Designer Tool

The ShapeDesigner tool that comes with JsDiagram lets you create and edit custom shapes interactively and save them to Shape Library files. When the tool starts, a new shape library is created by default with a single basic rectangular shape in it. To modify the shape, drag and drop shape elements from the palette to the right onto the shape canvas. The following drop operations are supported:

- dropping an element over existing element of a shape outline replaces the element (the element to be replaced will be marked in red)
- dropping an element over the joint of two outline elements inserts the element between them (the joint will be marked in blue)
- dropping an element on an empty area of the canvas adds a decoration to the Shape
- dropping a decoration over the shape outline replaces all elements of the outline with ones from the decoration
- dropping a decoration on an empty area of the canvas adds a filled ShapeDecoration to the **Shape**
- dropping a decoration over a **ShapeDecoration** replaces its elements

Elements/decorations can be further customized by dragging their control points.
You can delete, copy and paste dropped elements/decorations, as well as undo and redo drop actions via the options in the Edit menu.
Styling can be specified to elements/decorations by using the controls in the property panel, that is shown when an element/decoration is selected.
The Preview panel lets you see how your shape will look and behave in a real DiagramView control.
New shapes can be added from the Shape/New menu option.
The finished library can be exported to an .sl file from the File/Save Library menu option.

Animations

Item Animations

The animate method of DiagramItem can run following animations:

- FillColorAnimation animates the fill color of nodes or link arrowheads.
- StrokeColorAnimation animates the color of node borders or link segments.
- OpacityAnimation animates items' opacity for fade-in and fade-out effects.

The following animations can be applied to DiagramNodes:

- BounceAnimation animates vertical position of nodes.
- ShakeAnimation animates horizontal position of nodes.
- PulseAnimation animates node's local scale factor.
- PathAnimation moves nodes along specified path.

Item's animated values are considered transient and do not change actual property values. If you want to assign last value from animation to actual property such as bounds or opacity, set the *keepLastValue* flag in animation parameters:

JavaScript Copy Code

```
node.animate(
    new PathAnimation(
        [
            new Point(30, 30),
            new Point(50, 30),
            new Point(75, 66),
            new Point(125, 88)
        ],
        {
            keepLastValue: true
        }));
```

Other animation parameters include duration, repeat, repeatCount, from / to values, animation curve and easing options:

JavaScript Copy Code

```
node.animate(
    new OpacityAnimation(
        {
            fromValue: 1,
            toValue: 0.5,
            duration: 250,
            repeat: true,
            repeatCount: 4,
            reverse: true,
            animationType: Animations.AnimationType.BackBow
        }));
```

Item and view animations that affect different properties can run simultaneously.

View Animations

The animate method of DiagramView can run following animations:

- ScrollAnimation animates the scroll position of a view.

- ZoomAnimation animates the zoom factor of a view.

```
JavaScript                                                    Copy Code
diagramView.animate(
        new ScrollAnimation(
        {
            fromValue: new Point(0, 0),
            toValue: new Point(100, 60),
            animationType: Animations.AnimationType.Power,
            easingType: Animations.EasingType.EaseInOut
        }));
```

View animations are not transient and always set the scroll* or zoomFactor properties.

Looping Animations

To create looping animations, use the repeat and reverse properties. To stop an animation, call its stop method.

```
JavaScript                                                    Copy Code
// Use this property to repeat the animation until stopped.
animation.repeat = true;
// Use this property to reverse the to and from values.
animation.reverse = true;

// Use the stop method to complete the animation.
animation.stop();
```

Custom Animations

To animate arbitrary attributes of any type, an Animation instance can receive an **updateCallback** function as a parameter to the constructor. This callback function can be used to set item coordinates or property values of the object being animated. The callback signature should contain two arguments - the animation object instance and a value representing the animated property delta. This callback is called at every animation timer "tick". The following example creates a pulse-like effect, using an update callback:

```
JavaScript                                                    Copy Code
var Animation = MindFusion.Animations.Animation;
var Utils = MindFusion.Diagramming.Utils;

var options =
    {
        duration: 550,
        fromValue: -0.15,
        toValue: 0.15,
        repeat: true,
        reverse: true
    };

/**
* @param [Animation] animation The animation object instance.
* @param [Number] animationDelta A number representing the percentage of displacement
* of the animated property's value in the [fromValue, toValue] range.
*/
function pulse(animation, animationDelta)
{
    var bounds = animation.item.bounds.clone();
    var to = animation.toValue;
    bounds = Utils.inflate(bounds, to * animationDelta, to * animationDelta);
    animation.item.setBounds(bounds, true);
}

var animation = new Animation(node, options, pulse);
animation.start();
```

Alternatively, a class derived from **Animation** could override the virtual **onUpdate** method.

Default Animation and Easing Functions

There are seven default animation type functions that can be used to animate items, designated by respective elements of AnimationType enumeration:

- Linear: represents an animation that accelerates / decelerates using a linear function.

- Power: represents an animation that accelerates / decelerates through exponentiation formula - x^n where n can be set through the animation's param property. The default exponent value is 2.
- Exponential: represents an animation that accelerates / decelerates using an exponential function. The exponent can be set through the animation's **param** property. The default exponent value is 10.
- Circular: represents an animation that accelerates / decelerates using a circular function.
- BackBow: represents an effect that simulates a retraction in the motion of the animation. The amplitude of the retraction can be controlled through the animation's **param** property. The default param value is 1.5.
- Bounce: represents a bouncing effect animation.
- Elastic: represents an oscillating effect animation. The oscillation frequency can be controlled through the animation's **param** property. The default param value is 0.5.

The default animations can be additionally customized by an easing function, specified via EasingType enumeration:

- EaseIn: follows the animation function associated with the Animation.
- EaseOut: inverses the animation function associated with the **Animation**.
- EaseInOut: uses **EaseIn** for the first half of the **Animation**'s duration and **EaseOut** for the second half.
- EaseOutIn: uses **EaseOut** for the first half of the **Animation**'s duration and **EaseIn** for the second half.

Custom Animation Functions

In addition to the provided default animation functions, it is also possible to define custom animation effects. Set the animationType property to Custom and pass a custom animation function callback to the Animation constructor. The callback signature should contain two arguments - the time progressed as a percentage from the start of the animation and the user-set parameter, used in the animation function. The following example uses a custom function where x is a value between 0 and 1 and y is a parameter, passed as an argument to the **Animation** constructor.

JavaScript Copy Code

```
var Animation = MindFusion.Animations.Animation;
var AnimationType = MindFusion.Animations.AnimationType;

var options =
    {
        duration: 500,
        fromValue: 5,
        toValue: 100,
        param: 5,
        animationType: AnimationType.Custom
    };

/**
* @param [Number] progress The time progress in the animation loop. A value between 0 and 1.
* @param [Object] param The user-set value, used to parameterize the animation function.
*/
function custom(progress, param)
{
    return Math.pow(Math.E, 1 - (1 / Math.pow(progress, param)));
}

var animation = new Animation(node, options, null, custom);
animation.start();
```

.NET Core Assemblies

The .NET Core assemblies included in Diagramming / MVC distribution provide mostly the same API as ASP.NET MVC 5 builds of the library, with a few missing features and differences listed below:

JSON converters

Since ASP.NET Core does not provide JSON converters -based serialization, the diagramming assemblies for .NET Core do not include converter classes and associated properties and methods - RegisterConverters, ItemTypes, MaxJsonLength, RecursionLimit, SetMaxJsonLength, SetRecursionLimit. In order to serialize custom node types, you must override their SaveToJson and LoadFromJson methods instead of implementing converters, and call the RegisterItemType method or its overload taking composite node template as an argument. ItemTypeResolver does not have a base class and does not include ResolveType, ResolveTypeId, RegisterConverters, Converters and ItemTypes members.

Other differences

- DiagramViewExtensions methods return **IHtmlContent** instead of **MvcHtmlString**.
- Width and Height properties, SetWidth and SetHeight extension methods which have been declared obsolete in MVC 5 version are not available in .NET Core assemblies.

Initial Values of Item Properties

Each new item, created either by the user or programmatically, is initialized with some kind of default property values.

If you create items by using the **new** operator and adding them the diagram's Nodes or Links collection, you can initialize the items by passing a Diagram reference to the respective constructor, or by passing another item to the copy constructor. In the former case, item properties are initialized from the default-property values defined in **Diagram**. In the latter case, properties are copied from the prototype item.

When users draw new items, the items are initialized from the **Diagram** properties, but you can change the default values by handling the initializeNode and initializeLink events.

Items created via the diagram's Factory object are always initialized from the default-property values defined in **Diagram**.

Below are listed item properties and the corresponding constants or **Diagram** properties through which they are initialized.

Note that some properties in the tables below are marked as *styled*. Styled properties usually default to **null** (**Nothing** in Visual Basic), unless the item is created through the prototype constructor, in which case the values of the styled properties are obtained from the prototype item. When a style property is **null**, its effective value is determined by the style associated with the item or the theme applied to the diagram. For more information on how the style system works, check Styles and Themes.

DiagramItem Property	Initialized With
Brush (*styled*)	**null** (affected by Brush of the corresponding DiagramItemStyle)
Locked	**false**
Font (*styled*)	**null** (affected by FontFamily, FontSize, FontStyle, and FontUnit of the corresponding DiagramItemStyle)
Pen (*styled*)	**null** (affected by Stroke and StrokeThickness of the corresponding DiagramItemStyle)
ShadowBrush (*styled*)	**null** (affected by ShadowBrush of the corresponding DiagramItemStyle)
ShadowOffsetX	ShadowOffsetX
ShadowOffsetY	ShadowOffsetY
Tag	**null**
TextBrush (*styled*)	**null** (affected by TextBrush of the corresponding DiagramItemStyle)
ToolTip	(empty string)
Visible	**true**

ShapeNode Property	Initialized With
EnabledHandles	AdjustmentHandles.All
Expandable	NodesExpandable
ImageUrl	**(empty string)**
ImageAlign	ImageAlign.Fit
Text	ShapeText
TextFormat	TextFormat
Transparent	**false**

ContainerNode Property	Initialized With
Caption	ContainerCaption
CaptionHeight	ContainerCaptionHeight
Foldable	ContainersFoldable

Folded	**false**
Margin	ContainerMargin

TableNode Property	Initialized With
Caption	TableCaption
ConnectionStyle	TableConnectionStyle
Scrollable	TablesScrollable

DiagramLink Property	Initialized With
BaseShape	LinkBaseShape
HeadShape	LinkHeadShape
Shape	LinkShape
Text	LinkText

Tutorials

Tutorial 1: Getting started

This tutorial will show you how to set up your project to use MvcDiagram.

1. Creating an ASP.NET Core Application

Go to File->New->Project. From the list of installed templates, choose ASP.NET Core Web App (Model-View-Controller) and hit Next. Enter "Tutorial1" as name of the application and press Next. On the next screen just press Create.

2. Installing the package

From Solution Explorer menu -> Dependencies -> Manage NuGet Packages, install the MindFusion.Diagramming.Mvc package.

3. Referencing the client scripts

The client script *MindFusion.Diagramming.js* should appear automatically in the *wwwroot/js* subfolder once the project is built. If you need to change the location of the *MindFusion.Diagramming.js* script, set the ClientScriptLocation property to configure the control:

C# Copy Code

```
view.ClientScriptLocation = "/SomeFolder/MindFusion.Diagramming.js";
```

4. Adding a DiagramView to the page

Open the Index.cshtml page, which is located below the Views->Home node in the Solution Explorer tree. Add references to the MindFusion.Diagramming.Mvc namespace:

HTML Copy Code

```
@using MindFusion.Diagramming.Mvc
```

Add a call to the HTML helper's DiagramView method by typing the following line:

HTML Copy Code

```
@Html.DiagramView((DiagramView)ViewBag.DiagramView, new { style = "width:800px; height:700px;" })
```

5. Creating the diagram

Open the controller class HomeController.cs, which is located in the Controllers node in the Solution Explorer tree. On top add the following references:

C# Copy Code

```
using Microsoft.Maui.Graphics;
using MindFusion.Diagramming;
using MindFusion.Diagramming.Mvc;
using SolidBrush = MindFusion.Drawing.SolidBrush;
using LinearGradientBrush = MindFusion.Drawing.LinearGradientBrush;
```

In the body of the Index method add

C# Copy Code

```
DiagramView view = new DiagramView("diagramView1");
```

The code above creates a new DiagramView object with Id set to "diagramView1".

Change some properties of the view's Diagram object:

C# Copy Code

```
Diagram diagram = view.Diagram;
diagram.BackBrush = new SolidBrush(Colors.White);
diagram.LinkHeadShape = ArrowHeads.Triangle;
diagram.LinkHeadShapeSize = 3;

DiagramStyle style = new DiagramStyle();
style.FontFamily = "Verdana";
style.Brush = new LinearGradientBrush(Colors.LightBlue, Colors.Blue, 90);
style.Stroke = new SolidBrush(Colors.Black);
diagram.Style = style;
```

Diagram's BackBrush property defines the default color with which the interior of the diagram will be filled. Diagram's LinkHeadShape and LinkHeadShapeSize properties define the default base shape and size for links' ending arrowhead. Additional styles can be applied to the diagram items by using the diagram's Style property.

Add the following code to add some ShapeNodes to the diagram. ShapeNode objects can be drawn as various geometric shapes, as set through their Shape property. There are over 100 predefined shapes exposed as static members of the Shapes class. The **Shape** property can be set to one of the predefined shapes: RoundRect, Rectangle, Ellipse, Process, Procedure, Delay and much more.

C# Copy Code

```
// add some nodes and links to the view's Diagram
ShapeNode node1 = diagram.Factory.CreateShapeNode(new Rect(55, 10, 40, 15));
node1.Shape = Shapes.Terminator;
node1.Text = "Start";
node1.Brush = new LinearGradientBrush(Colors.Yellow, Colors.Orange, 90);

ShapeNode node2 = diagram.Factory.CreateShapeNode(new Rect(50, 40, 50, 20));
node2.Text = "Receive order via e-mail";

ShapeNode node3 = diagram.Factory.CreateShapeNode(new Rect(50, 70, 50, 20));
node3.Text = "Copy and paste e-mail data into database";

ShapeNode node4 = diagram.Factory.CreateShapeNode(new Rect(55, 100, 40, 40));
node4.Shape = Shape.FromId("Decision");
node4.Text = "Shipping involved?";
node4.Brush = new LinearGradientBrush(Colors.White, Colors.Fuchsia, 90);

ShapeNode node5 = diagram.Factory.CreateShapeNode(new Rect(100, 135, 50, 20));
node5.Shape = Shape.FromId("Save");
node5.Text = "Print invoice and UPS label";
node5.Brush = new SolidBrush(Colors.Chartreuse);

ShapeNode node6 = diagram.Factory.CreateShapeNode(new Rect(100, 165, 50, 20));
node6.Text = "Send e-mail to confirm shipping";
diagram.Nodes.Add(node6);

ShapeNode node7 = diagram.Factory.CreateShapeNode(new Rect(100, 195, 50, 20));
node7.Text = "Assemble package and ship";

ShapeNode node8 = diagram.Factory.CreateShapeNode(new Rect(55, 230, 40, 15));
node8.Shape = Shapes.Terminator;
node8.Text = "End";
node8.Brush = new LinearGradientBrush(Colors.Yellow, Colors.Orange, 90);
```

Add the following code to add some DiagramLinks to the diagram:

C# Copy Code

```
diagram.Factory.CreateDiagramLink(node1, node2);
diagram.Factory.CreateDiagramLink(node2, node3);
diagram.Factory.CreateDiagramLink(node3, node4);

DiagramLink link4 = diagram.Factory.CreateDiagramLink(node4, node5);
link4.Text = "Yes";

diagram.Factory.CreateDiagramLink(node5, node6);
diagram.Factory.CreateDiagramLink(node6, node7);
diagram.Factory.CreateDiagramLink(node7, node8);

DiagramLink link8 = diagram.Factory.CreateDiagramLink(node4, node8);
link8.Text = "No";
```

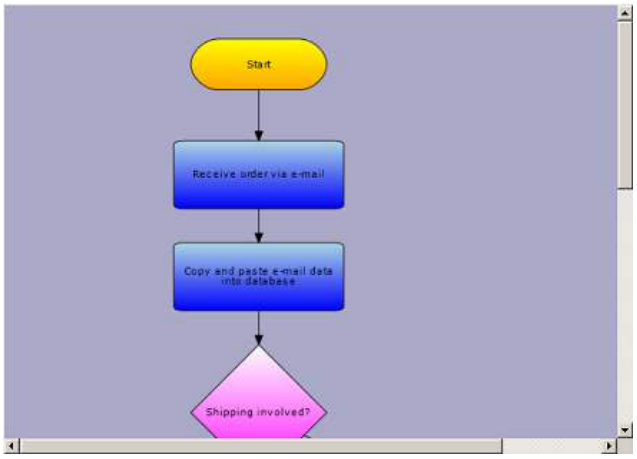
Finally, store the newly created view in the **ViewBag** dictionary:

C# Copy Code

```
ViewBag.DiagramView = view;
```

6. Running the application

From the main menu choose Debug->Start without debugging or press CTRL+F5. The image below depicts how the application would look like:



Tutorial 2: Handling client-side events

This tutorial will show you how to use client events in MvcDiagram.

1. Create a new ASP.NET Core MVC Application and add a DiagramView control to the page

Follow steps 1-4 from Tutorial 1: Getting started.

2. Create and arrange the diagram

Open the HomeController.cs file and add a reference to the MindFusion.Diagramming.Layout namespace in the 'using' section of the page:

C# Copy Code

```
using MindFusion.Diagramming.Layout;
```

Add the following code block to create a new DiagramView object and add some nodes and links to its Diagram.

C# Copy Code

```
// initialize the DiagramView
DiagramView view = new DiagramView("diagramView1");
```

```
Diagram diagram = view.Diagram;

// add some nodes and links
for (var i = 0; i < 5; i++)
{
    var node = diagram.Factory.CreateShapeNode(new Rect(0, 0, 25, 25));
    node.Text = "node " + i;
    diagram.Nodes.Add(node);
    if (i > 0)
    {
        var link = diagram.Factory.CreateDiagramLink(diagram.Nodes[0], node);
        diagram.Links.Add(link);
    }
}
```

Set the Brush property of the first node to DarkRed and its Text to "Root":

C# Copy Code

```
var rootNode = diagram.Nodes[0];
rootNode.Brush = new SolidBrush(Colors.DarkRed);
((ShapeNode)rootNode).Text = "Root";
```

Create a FractalLayout object and apply it to the Diagram:

C# Copy Code

```
var layout = new FractalLayout();
layout.Root = rootNode;
layout.Arrange(diagram);
```

Add the newly created view to the **ViewData** dictionary:

C# Copy Code

```
ViewBag.DiagramView = view;
```

Modify the call to the DiagramView HTML helper method as following to set the NodeCreatedScript to "nodeCreated" and NodeDeletedScript to "nodeDeleted". These are the names of the JavaScript functions that will handle the corresponding events. We will define them a bit later in the *Index.cshtml* file.

HTML Copy Code

```
@Html.DiagramView(
    ((DiagramView)ViewBag.DiagramView)
    .NodeCreatedScript("nodeCreated")
    .NodeDeletedScript("nodeDeleted"),
    new { style="width:700px; height:500px;" }
)
```

3. Handling client events

Open the site.js file, located in the wwwroot/js folder and add the following JavaScript block:

JavaScript Copy Code

```
function nodeCreated(sender, args)
{
    if (args.node)
    {
        // send an ajax request to the Controller's Test action method
        $.ajax(
        {
            type: "POST", url: "/Home/Test",
            data: { json: sender.toJson(), index: sender.nodes.indexOf(args.node) },
            dataType: 'json',
            success: function (result) {
                sender.fromJson(result.success);
            },
            error: function (req, status, error) {
                alert(error);
            }
        });
    }
```

```
}

function nodeDeleted(sender, args)
{
    if (args.node.zIndex == 0)
    {
        if (sender.nodes.length > 0)
        {
            // set the root node to the first node in the diagram.nodes collection
            var rootNode = sender.nodes[0];
            rootNode.brush = { type: 'SolidBrush', color: 'firebrick' };
            rootNode.text = "root";
            rootNode.zIndex = 0;

            // create links between the root and children nodes
            for (var i = 1; i < sender.nodes.length; i++)
            {
                sender.factory.createDiagramLink(
                    sender.nodes[0], sender.nodes[i]);
            }
        }
    }
}
```

Open the *HomeController.cs* file and add the Test action method:

C# Copy Code

```
public JsonResult Test(string json, string index)
{
    DiagramView view = DiagramView.FromJson(json);
    Diagram diagram = view.Diagram;
    var newNode = ((ShapeNode)diagram.Nodes[Int32.Parse(index)]);
    newNode.Text = "node " + (diagram.Nodes.Count-1).ToString();
    if (diagram.Nodes.Count > 1)
    {
        // create a link between the root node and the newly created node and apply the layout to
the diagram
        var link = diagram.Factory.CreateDiagramLink(diagram.Nodes[0], newNode);
        var layout = new FractalLayout();
        layout.Root = diagram.Nodes[0];
        layout.Arrange(diagram);
    }
    else
    {
        // set the newly created node as a root node
        newNode.Brush = new SolidBrush(Colors.DarkRed);
        ((ShapeNode)newNode).Text = "Root";
    }
    return Json(new { success = DiagramView.ToJson(view) });
}
```

When a new node is added to the diagram the **nodeCreated** function will be called. It will send an ajax request, containing the serialized to JSON view and the index of the new node to the controller's Test action method. Within the Test method the view will be recreated from the JSON string and depending of the number of nodes in the diagram either a link between the root node and the new node will be created or the new node will be set as root.

The **nodeDeleted** function is called upon node deletion. If the deleted node is the root node a new root node will be assigned and will be linked to all other nodes in the diagram.

4. Run the application

Run the application. Try adding and deleting some nodes to see how the events will be handled.

Tutorial 3: Overview and NodeListView

This tutorial shows how to use the Overview and NodeListView components.

1. Create a new ASP.NET Core MVC Application and add a

DiagramView control to the page

Follow steps 1-5 from Tutorial 1: Getting started.

2. Add the controls to the page

Add the following initialization code for the DiagramView, Overview, and NodeListView controls in the Index method of the HomeController.cs:

HTML Copy Code

```
diagram.ShowGrid = true;
diagram.GridColor = Colors.Red;
diagram.AlignToGrid = true;
view.AllowInplaceEdit = true;

Overview overview = new Overview("overview1", "diagramView1");
overview.BackColor = Colors.LightGray;
overview.DimColor = Colors.White;
ViewBag.Overview = overview;

NodeListView nodelist = new NodeListView("nodelist1");
for (var i = 0; i < Shape.Shapes.Count; ++i)
{
    if (Shape.Shapes[i].Id != "RoundRect1" && !Shape.Shapes[i].IsArrowhead)
    {
        var node = new MindFusion.Diagramming.ShapeNode();
        node.Brush = new MindFusion.Drawing.SolidBrush(Colors.LightBlue);
        node.Shape = Shape.Shapes[i];
        nodelist.AddNode(node);
    }
}
ViewBag.NodeListView = nodelist;
```

Setting the Diagram's ShowGrid property to true will display the alignment grid, and setting the AlignToGrid property will ensure the that the sides newly created nodes will be aligned to the nearest grid point.

The view's AllowInplaceEdit property defines whether items' text can be edited inplace.

The **Overview**'s BackColor property defines the background color of the control and the DimColor property sets the color of the displayed diagram.

Use the **NodeListView** control's AddNode method to add an arbitrary number of nodes to its nodes collection.

Remove the call to the Diagram's HTML Helper method and add the following form declaration instead:

HTML Copy Code

```
@using (Html.BeginForm())
{
    <input type="submit" name="submitButton" value="submit" />

    <div style="display:flex;">
        @Html.DiagramView(((DiagramView)ViewBag.DiagramView), new { style="width:700px;
height:700px;" })
        @Html.Overview(((Overview)ViewBag.Overview), new { style="width:200px; height:200px;" })
        @Html.NodeListView(((NodeListView)ViewBag.NodeListView), new { style="width:200px;
height:500px;" })
    </div>
}
```

3. Submitting the form

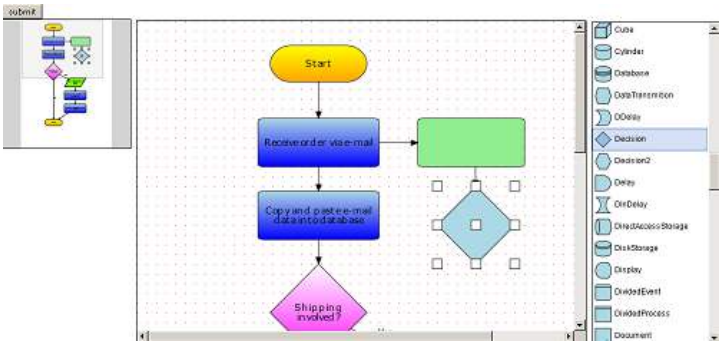
Upon form submission the DiagramView control will serialize itself in JSON format and will store the serialized string in a hidden field. This field will be added automatically to the form by the **DiagramView** control and will have a name matching the following pattern: DiagramViewId + "_PostData" e.g. "diagramView1_PostData". You can access the field through the **Request.Form** collection and pass the data to the DiagramView's FromJson method in order to keep the changes made from the client. The view's FromRequest static method can be used as a shortcut to achieve the same effect. Change the **DiagramView** control initializer as follows to persist the changes:

C# Copy Code

```
DiagramView view = DiagramView.FromRequest("diagramView1", Request);
```

4. Running the application

Run the application. Try drawing some nodes and/or links. Double-click on an object to inplace-edit it's text. Drag items from the **NodeListView** control onto the **Diagram** to create new nodes. Submit the form to see how the control will store your changes. The image below depicts how the application would look like with some nodes and links added to the diagram.



Tutorial 4: Loading graph data

This tutorial shows how to load graph data from XML and create diagram objects corresponding to the graph nodes and edges.

1. Create a new ASP.NET MVC Application and add a DiagramView control to the page

Follow steps 1-5 from Tutorial 1: Getting started.

2. Create the XML file

From the project's context menu choose Add -> New Item. Create a new XML file called *SampleGraph.xml* and add the following markup to its content.

XML Copy Code

```
<?xml version="1.0" encoding="utf-8" ?>
<Graph>
    <Nodes>
        <Node id="0" name="start" />
        <Node id="1" name="activity 1" />
        <Node id="2" name="task 1" />
        <Node id="3" name="task 2" />
        <Node id="4" name="activity 2" />
        <Node id="5" name="task 3" />
        <Node id="6" name="task 4" />
        <Node id="7" name="activity 3" />
        <Node id="8" name="task 5" />
        <Node id="9" name="task 6" />
        <Node id="10" name="end" />
    </Nodes>
    <Links>
        <Link origin="0" target="1" />
        <Link origin="1" target="2" />
        <Link origin="1" target="3" />
        <Link origin="2" target="4" />
        <Link origin="3" target="4" />
        <Link origin="4" target="5" />
        <Link origin="4" target="6" />
        <Link origin="5" target="10" />
        <Link origin="6" target="10" />
        <Link origin="0" target="7" />
        <Link origin="7" target="8" />
        <Link origin="8" target="9" />
        <Link origin="1" target="9" />
        <Link origin="9" target="10" />
    </Links>
</Graph>
```

Move the xml file to the wwwroot/js subfolder. Set the file's "Build Action" property to "Content" and "Copy to Output Directory" to "Copy if newer".

3. Load the XML document

Open HomeController.cs file and and the following references

C# Copy Code

```
using Microsoft.AspNetCore.Hosting;
using System.Xml.Linq;
using MindFusion.Diagramming.Layout;
```

We will get access to the SampleGraph.xml file, located in the wwwroot/js folder by injecting and querying the **IHostingEnvironment** service. Add a private variable
and a constructor method to the HomeController class as follows:

C# Copy Code

```
public class HomeController : Controller
{
    private readonly IWebHostEnvironment env;
    public HomeController(IWebHostEnvironment hostEnvironment)
    {
        env = hostEnvironment;
    }
```

Delete the code for creating nodes and links in the code block and add declarations for the following variables. "nodeMap" maps diagram nodes to their identifiers, and "bounds" contains the default node size.

C# Copy Code

```
Dictionary<string, DiagramNode> nodeMap = new Dictionary<string, DiagramNode>();
Rect bounds = new Rect(0, 0, 60, 20);
```

Load the XML document and its root element using the **Linq for XML** API:

C# Copy Code

```
var path = env.WebRootFileProvider.GetFileInfo("js//SampleGraph.xml")?.PhysicalPath;
var xml = XDocument.Load(path);
var graph = xml.Element("Graph");
```

4. Create the Shape nodes

Load the graph node elements and create their corresponding ShapeNode objects by calling the CreateShapeNode method of the Factory class. **CreateShapeNode** is just a shortcut to creating a **ShapeNode** instance using the "new" operator and adding it to the Nodes collection of the diagram. Additionally, this code maps the new node to the "id" attribute of the XML element, and sets the node's Text to the value of the "name" attribute.

C# Copy Code

```
var nodes = xml.Descendants("Node");
foreach (var node in nodes)
{
    var diagramNode = diagram.Factory.CreateShapeNode(bounds);
    nodeMap[node.Attribute("id").Value] = diagramNode;
    diagramNode.Text = node.Attribute("name").Value;
}
```

5. Create the Diagram links

Load the graph links and create corresponding DiagramLink objects. The Origin and Destination of the links are accessed by their ids through the "nodeMap" dictionary.

C# Copy Code

```
var links = xml.Descendants("Link");
foreach (var link in links)
{
    diagram.Factory.CreateDiagramLink(
    nodeMap[link.Attribute("origin").Value],
```

```
    nodeMap[link.Attribute("target").Value]);
}
```

6. Arrange the diagram

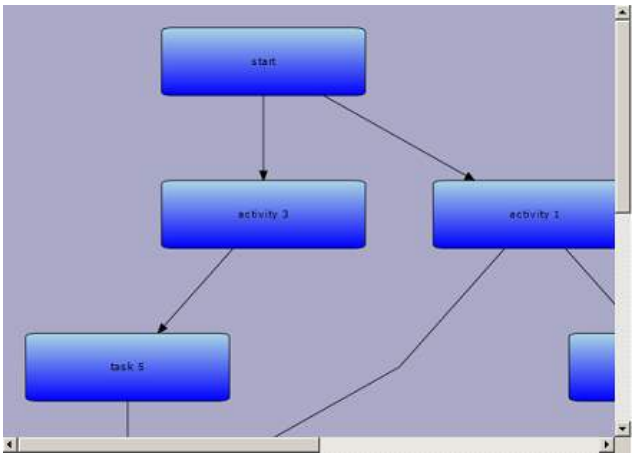
Arrange the diagram using the LayeredLayout class. Call the Diagram's ResizeToFitItems method to ensure that its bounds will fit all diagram items.

C# Copy Code

```
var layout = new LayeredLayout();
layout.NodeDistance = 80;
layout.Arrange(diagram);
diagram.ResizeToFitItems(5);
```

7. Run the application

Build and run the project. If everything is fine, you should see this representation of the graph:



Tutorial 5: Loading hierarhical data

This tutorial shows how to load hierarchical data recursively from XML and create diagram items corresponding to the hierarchy.

1. Create a new ASP.NET MVC Application and add a DiagramView control to the page

Follow steps 1-4 from Tutorial 1: Getting started.

2. Create the XML file

From the project's context menu choose Add -> New Item. Create a new XML file called *SampleTree.xml* and add the following markup to its content.

XML Copy Code

```
<?xml version="1.0" encoding="utf-8" ?>
<Project>
    <Activity Name="Activity 1">
        <Activity  Name="sub-activity 1">
            <Activity  Name="sub-activity 1-1" />
            <Activity  Name="sub-activity 1-2" />
        </Activity>
        <Activity  Name="sub-activity 2">
            <Activity  Name="sub-activity 2-1" />
            <Activity  Name="sub-activity 2-2" />
            <Activity  Name="sub-activity 2-3" />
        </Activity>
        <Activity  Name="sub-activity 3">
            <Activity  Name="sub-activity 3-1" />
            <Activity  Name="sub-activity 3-2" />
        </Activity>
    </Activity>
```

```
    <Activity Name="Activity 2">
        <Activity  Name="sub-activity 1">
            <Activity  Name="sub-activity 1-1" />
            <Activity  Name="sub-activity 1-2" />
            <Activity  Name="sub-activity 1-3" />
        </Activity>
        <Activity  Name="sub-activity 2">
            <Activity  Name="sub-activity 2-1" />
            <Activity  Name="sub-activity 2-2" />
        </Activity>
    </Activity>
    <Activity Name="Activity 3">
        <Activity  Name="sub-activity 1">
            <Activity  Name="sub-activity 1-1" />
            <Activity  Name="sub-activity 1-2" />
            <Activity  Name="sub-activity 1-3" />
            <Activity  Name="sub-activity 1-4" />
        </Activity>
        <Activity  Name="sub-activity 2" />
        <Activity  Name="sub-activity 3">
            <Activity  Name="sub-activity 3-1" />
            <Activity  Name="sub-activity 3-2" />
        </Activity>
    </Activity>
</Project>
```

Move the xml file to the wwwroot/js subfolder. Set the file's "Build Action" property to "Content" and "Copy to Output Directory" to "Copy if newer".

3. Load the XML document

Open *HomeController.cs* file and and the following references:

C# Copy Code

```
using System.Xml.Linq;
using Microsoft.Maui.Graphics;
using MindFusion.Diagramming;
using MindFusion.Diagramming.Mvc;
using MindFusion.Diagramming.Layout;
using SolidBrush = MindFusion.Drawing.SolidBrush;
using LinearGradientBrush = MindFusion.Drawing.LinearGradientBrush;
```

Add the following member to the page class to hold the default node size:

C# Copy Code

```
var nodeBounds = new Rect(0, 0, 30, 10);
```

Add the diagram's initialization code to the body of the Index method.

C# Copy Code

```
DiagramView view = new DiagramView("diagramView1");

Diagram diagram = view.Diagram;
diagram.LinkHeadShape = ArrowHeads.Triangle;
diagram.LinkHeadShapeSize = 3;

DiagramStyle style = new DiagramStyle();
style.FontFamily = "Verdana";
style.BackBrush = new SolidBrush(Colors.White);
style.Brush = new LinearGradientBrush(Colors.LightBlue, Colors.Blue, 90);
style.Stroke = new SolidBrush(Colors.Black);
diagram.Style = style;

ViewBag.DiagramView = view;
return View();
```

4. Define CreateChildren method

Define *CreateChildren* as follows. It takes as parameters the parent DiagramNode and parent XML element, iterates the child XML elements to build the next level of the hierarchy, and creates corresponding diagram items by calling CreateShapeNode and CreateDiagramLink.

C# Copy Code

```
private void CreateChildren(Diagram diagram, DiagramNode parentDiagNode, XElement
parentXmlElement)
{
    foreach (var element in parentXmlElement.Elements("Activity"))
    {
        var node = diagram.Factory.CreateShapeNode(nodeBounds);
        node.Text = element.Attribute("Name").Value;
        diagram.Factory.CreateDiagramLink(parentDiagNode, node);
        CreateChildren(diagram, node, element);
    }
}
```

Add the following block to the Index method to create the tree root node, and add a call to the CreateChildren method, which builds the tree recursively.

C# Copy Code

```
var root = diagram.Factory.CreateShapeNode(nodeBounds);
root.Text = "Project";
var path = env.WebRootFileProvider.GetFileInfo("js//SampleTree.xml")?.PhysicalPath;
var document = XDocument.Load(path);
CreateChildren(diagram, root, document.Element("Project"));
```

5. Arrange the diagram

Use TreeLayout to arrange the diagram. Add the following code to the end of the Index method.

C# Copy Code

```
var layout = new TreeLayout();
layout.Type = TreeLayoutType.Cascading;
layout.Direction = TreeLayoutDirections.LeftToRight;
layout.LinkStyle = TreeLayoutLinkType.Cascading2;
layout.NodeDistance = 5;
layout.LevelDistance = -25; // let horizontal positions overlap
layout.Arrange(diagram);
```

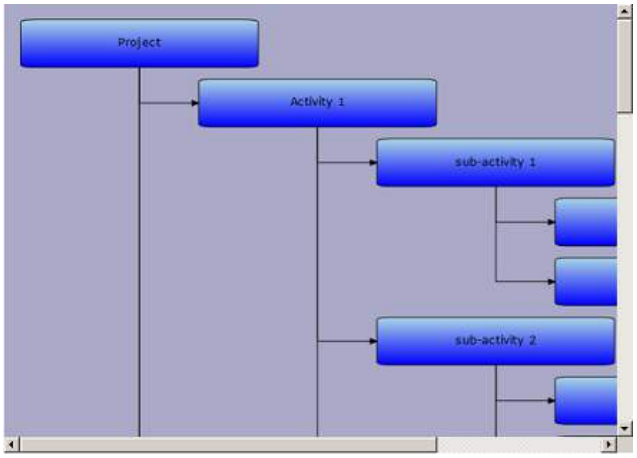
Call the Diagram's ResizeToFitItems method to ensure that its bounds will fit all diagram items. Set the ZoomFactor property of the view to 70.

C# Copy Code

```
diagram.ResizeToFitItems(5);
```

6. Run the application

Build and run the project. If everything is fine, you should see this representation of the hierarchy:



Tutorial 6: Create a Custom Node Type

This tutorial shows how to define a custom node class that adds several new properties.

1. Create a new ASP.NET Core MVC Application and add a DiagramView control to the page

Follow steps 1-4 from Tutorial 1: Getting started.

2. Create the custom node type

Create a new js file in the wwwroot/js folder named OrgChartNode.js with the following content:

JavaScript — Copy Code

```
class OrgChartNode extends MindFusion.Diagramming.ShapeNode
{
    constructor(parent)
    {
        super(parent);

        this.title = 'title';
        this.fullName = 'full name';
        this.nodeImageLocation = '../images/icon4.png';

        this.shape = 'Rectangle';
        this.text = 'node text';
        this.brush = '#fff';
    }

    // drawing logic
    updateCanvasElements()
    {
        super.updateCanvasElements();

        // add the node image;
        if (this.nodeImage)
        {
            this.nodeImage.bounds = new Rect(
                this.bounds.x, this.bounds.y + (this.bounds.height - 20) / 2, 20, 20);
            this.graphicsContainer.content.push(this.nodeImage);
        }

        // add the title label
        var titleFont = Font.copy(this.effectiveFont);
        var titleLabel = new Text(this.title,
            new Rect(this.bounds.x + 21, this.bounds.y + 5, this.bounds.width - 22,
titleFont.size));
        titleLabel.font = titleFont;
        titleLabel.font.fontStyle = MindFusion.Drawing.FontStyle.Bold;
        titleLabel.fitInBounds = false;
        this.graphicsContainer.content.push(titleLabel);

        // add the name label
        var nameFont = Font.copy(this.effectiveFont);
        var nameLabel = new Text(this.fullName, new Rect(
            this.bounds.x + 21, this.bounds.y + 5 + titleFont.size, this.bounds.width - 22,
nameFont.size));
        nameLabel.font = nameFont;
        nameLabel.pen = "blue";
        nameLabel.fitInBounds = false;
        this.graphicsContainer.content.push(nameLabel);

        // adjust the text label properties
        var textFont = Font.copy(this.effectiveFont);
        textFont.size = 3;
        var textRect = new Rect(
            this.bounds.x + 21, this.bounds.y + 3 + titleFont.size + nameFont.size,
            this.bounds.width - 22, this.bounds.height - 2 - titleFont.size - nameFont.size);
        this._text.font = textFont;
        this._text.textAlignment = Alignment.Near;
        this._text.lineAlignment = Alignment.Near;
```

```
        this._text.margin = new MindFusion.Drawing.Thickness(0, 0, 0, 0);
        this._text.bounds = textRect;
    }

    loadNodeImage()
    {
        this.nodeImage.loaded = true;
        if (this.parent)
            this.invalidate();
    }

    // properties
    get title()
    {
        return this._title;
    }

    set title(value)
    {
        if (this._title !== value)
        {
            this._title = value;
        }
    }

    get fullName()
    {
        return this._fullName;
    }

    set fullName(value)
    {
        if (this._fullName !== value)
        {
            this._fullName = value;
        }
    }

    get nodeImageLocation()
    {
        return this._nodeImageLocation;
    }

    set nodeImageLocation(value)
    {
        if (this._nodeImageLocation != value)
        {
            this._nodeImageLocation = value;
            if (value)
            {
                this.nodeImage = new MindFusion.Drawing.Image(
                    new Rect(this.bounds.x, this.bounds.y + (this.bounds.height - 20) / 2, 20,
20));

                this.nodeImage.image.addEventListener("load", () => { this.loadNodeImage(); })
                this.nodeImage.image.src = value;
            }
        }
    }
}
```

The code above declares a new class called **OrgChartNode**, inheriting it from ShapeNode. It adds getters and setters for custom title, full name and image properties. **OrgChartNode** also exposes an updateCanvasElements override with the custom draw logic.

Add a reference to the new file to the end of the Index.cshtml:

HTML Copy Code

```
@section scripts {
    <script type="text/javascript" src="~/js/OrgChartNode.js"></script>
}
```

3. Register the custom type and create some test data

Modify the DiagramView declaration to set the value of the ControlLoadedScript property.

HTML

```
@Html.DiagramView(
    ((DiagramView)ViewBag.DiagramView).ControlLoadedScript("onLoaded"),
    new { style = "width:700px;height:500px;"})
```

Add the **onLoaded** function to *site.js*. The below code registers the **OrgChartNode** type, enables the interactive drawing of custom nodes and adds a couple of **OrgChartNode** instances to the diagram.

JavaScript

```
var Diagram = MindFusion.Diagramming.Diagram;
var Behavior = MindFusion.Diagramming.Behavior;
var Alignment = MindFusion.Diagramming.Alignment;

var Font = MindFusion.Drawing.Font;
var Rect = MindFusion.Drawing.Rect;
var Text = MindFusion.Drawing.Text;

var diagram = null;

function onLoaded(sender) {
    Diagram.registerClass(OrgChartNode, "OrgChartNode");

    diagram = sender.diagram;

    // enable drawing of custom nodes interactively
    diagram.customNodeType = OrgChartNode;
    sender.behavior = Behavior.Custom;

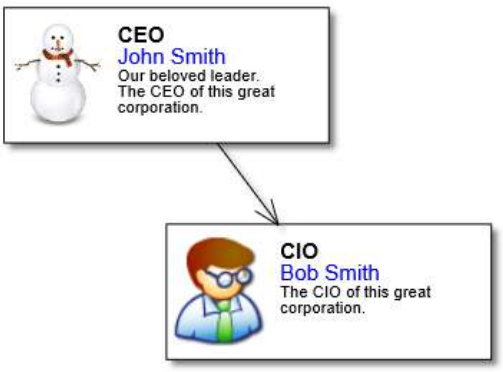
    var node1 = new OrgChartNode(diagram);
    node1.bounds = new Rect(25, 15, 60, 25);
    node1.title = "CEO";
    node1.fullName = "John Smith";
    node1.text =
        "Our beloved leader. \r\n" +
        "The CEO of this great corporation.";
    node1.nodeImageLocation = "../images/icon4.png";
    diagram.addItem(node1);

    var node2 = new OrgChartNode(diagram);
    node2.bounds = new Rect(55, 55, 60, 25);
    node2.title = "CIO";
    node2.fullName = "Bob Smith";
    node2.text = "The CIO of this great corporation.";
    node2.nodeImageLocation = "../images/icon5.png";
    diagram.addItem(node2);

    diagram.factory.createDiagramLink(node1, node2);
}
```

4. Running the application

Build and run. The image below depicts how the application would look like:



Tutorial 7: Custom Nodes Integration

This tutorial shows how to fully integrate the custom type with other MvcDiagram features, such as clipboard support, undo support, creating nodes by drag-and-drop or drawing with the mouse on the canvas.

1. Create a custom node type

Follow steps 1-3 from Tutorial 6: Create a Custom Node Type.

2. Modify OrgChartNode class

Add the following methods to the **OrgChartNode**'s declaration:

JavaScript — Copy Code

```
// support for the NodeListView drag'n'drop
clone()
{
    Diagram.suppressSetDirty = true;
    var copy = super.clone();

    copy.title = this.title;
    copy.fullName = this.fullName;
    copy.nodeImageLocation = this.nodeImageLocation;

    Diagram.suppressSetDirty = false;
    return copy;
}

// clipboard support
toJson()
{
    var json = super.toJson();

    json.title = this.title;
    json.fullName = this.fullName;
    json.nodeImageLocation = this.nodeImageLocation;

    return json;
}

fromJson(json)
{
    super.fromJson(json);

    this.title = json.title;
    this.fullName = json.fullName;
    this.nodeImageLocation = json.nodeImageLocation;
}

// undo/redo
saveState()
{
    var state = super.saveState();
```

```
    state.title = this.title;
    state.fullName = this.fullName;
    state.nodeImageLocation = this.nodeImageLocation;

    return state;
}

restoreState(state)
{
    super.restoreState(state);

    this.title = state.title;
    this.fullName = state.fullName;
    this.nodeImageLocation = state.nodeImageLocation;
}
```

3. Add a NodeListView control

Add a **NodeListView** control to the page:

HTML Copy Code

```
@Html.NodeListView(
    ((NodeListView)ViewBag.NodeListView),
    new { style = "width:300px;height:500px;"})
```

Add the initialization code for the **NodeListView** to the **onLoaded** handler:

JavaScript Copy Code

```
var Size = MindFusion.Drawing.Size;
...
nodeListView = MindFusion.Diagramming.NodeListView.find("nodeListView1");
nodeListView.measureUnit = 6;
nodeListView.padding = 1;
nodeListView.iconSize = new Size(50, 25);
nodeListView.defaultNodeSize = new Size(50, 25);

nodeListView.addNode(node1.clone());
nodeListView.addNode(node2.clone());
```

4. Enable the undo/redo support

Set the diagram's undoEnabled property value to true in the **onLoaded** handler:

JavaScript Copy Code

```
// enable undo/redo support
diagram.undoEnabled = true;
```

5. Call diagram methods

In *site.js*, add a global variable that will hold a reference to the diagramView object, so we can call its methods from the *Index* page. Set its value in the *onLoaded* handler.

JavaScript Copy Code

```
var diagramView = null;
...
diagramView = sender;
```

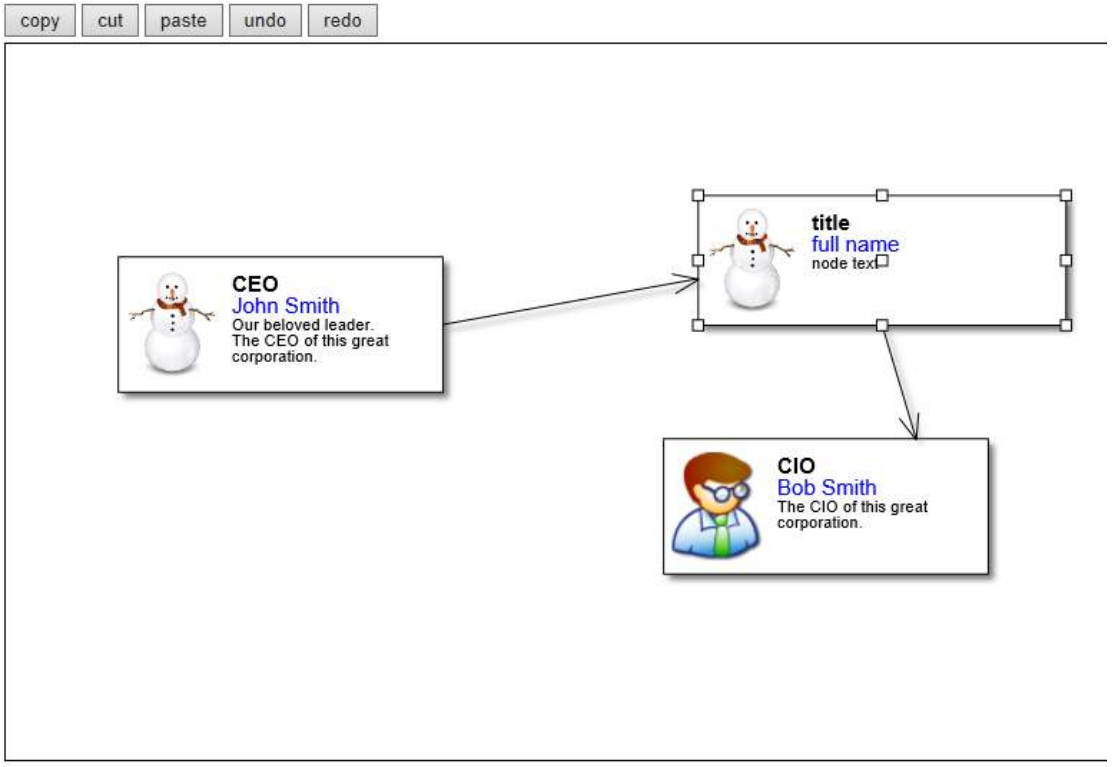
Add five buttons to the page - copy, cut, paste, undo and redo and define click handlers that will be used to call the corresponding diagram methods.

HTML Copy Code

```
<button onclick="diagramView.copyToClipboard()">copy</button>
<button onclick="diagramView.cutToClipboard()">cut</button>
<button onclick="diagramView.pasteFromClipboard(5, 5)">paste</button>
<button onclick="diagram.undo()">undo</button>
<button onclick="diagram.redo()">redo</button>
```

6. Running the application

Build and run. The image below depicts how the application would look like after adding some nodes and links:



Tutorial 8: Custom Composite Nodes

1. Create a new ASP.NET MVC Application and add a DiagramView control to the page. Follow steps 1-4 from Tutorial 1: Getting started. Add a NodeListView control to the page:

C# Copy Code

```
@Html.NodeListView(((NodeListView)ViewBag.NodeListView), new { style = "width:300px;height:500px;"
})
```

2. In the HomeController.cs file define a node template string that lists layout containers and graphics primitives from which nodes should be composed.

C# Copy Code

```
string template = @"{
 ""component"": ""SimplePanel"",
 ""children"":
 [
  {
   ""component"": ""Shape"",
   ""name"": ""Frame"",
   ""pen"": ""black"",
   ""brush"": ""white"",
   ""strokeThickness"": 0.2,
   ""isOutline"": true,
   ""autoProperty"": true
  },
 {
  ""component"": ""GridPanel"",
  ""rowDefinitions"": [""*""],
  ""columnDefinitions"": [""22"", ""*""],
  ""children"":
  [
   {
    ""component"": ""Image"",
    ""name"": ""Image"",
    ""autoProperty"": true,
```

```
    ""location"": ""../Content/icon4.png"",
    ""margin"": ""1"",
    ""imageAlign"": ""Fit""
   },
   {
    ""component"": ""StackPanel"",
    ""orientation"": ""Vertical"",
    ""gridColumn"": 1,
    ""margin"": ""1"",
    ""verticalAlignment"": ""Near"",
    ""children"":
    [
     {
      ""component"": ""Text"",
      ""name"": ""Title"",
      ""autoProperty"": true,
      ""text"": ""title"",
      ""font"": ""Arial bold""
     },
     {
      ""component"": ""Text"",
      ""name"": ""FullName"",
      ""autoProperty"": true,
      ""text"": ""full name"",
      ""pen"": ""blue"",
      ""padding"": ""1,0,1,0"",
      ""enableStyledText"": true
     },
     {
      ""component"": ""Text"",
      ""name"": ""Details"",
      ""autoProperty"": true,
      ""text"": ""details"",
      ""font"": ""Arial 3""
     }
    ]
   }
  ]
}]}
";
```

3. Add a new OrgChartNode class to the project and implement it as below. Its instances will represent the custom nodes posted back fro client side. The AutoJson attribute generates automatic JSON serialization .NET code to transfer values of properties defined in the template.

C# Copy Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

using MindFusion.Diagramming;
using MindFusion.Diagramming.Mvc;
using MindFusion.Json;

namespace Tutorial8.Models
{
    public class OrgChartNode : CompositeNode
    {
        public OrgChartNode()
        {
            Frame = "RoundRect";
        }

        public OrgChartNode(Diagram diagram)
            : base(diagram)
        {
            Frame = "RoundRect";
        }

        public OrgChartNode(OrgChartNode prototype): base(prototype)
        {
            Image = prototype.Image;
```

```
            Title = prototype.Title;
            FullName = prototype.FullName;
            Details = prototype.Details;
            Frame = prototype.Frame;
        }

        [AutoJson]
        public string Image { get; set; }

        [AutoJson]
        public string Title { get; set; }

        [AutoJson]
        public string FullName { get; set; }

        [AutoJson]
        public string Details { get; set; }

        [AutoJson]
        public string Frame { get; set; }
    }
}
```

4. Switch back to HomeController.cs. Set up the **DiagramView** and **NodeListView** controls. Custom item types registry is stored in the session. Call RegisterItemType once to register the custom class, and RegisterSession method for any control that will use the custom item type.

C# Copy Code

```
DiagramView view = new DiagramView("diagramView1");
view.Behavior = Behavior.Custom;
view.SetCustomNodeType(typeof(OrgChartNode));
view.Diagram.UndoManager.UndoEnabled = true;

// register current session
view.RegisterSession(HttpContext.Session);

// register custom item type
view.RegisterItemType(typeof(OrgChartNode), "OrgChartNode", "OrgChartNode", 100, template);

NodeListView nodeView = new NodeListView("nodeListView1");
nodeView.NodeSize =
nodeView.DefaultNodeSize = new Size(60, 25);

// register current session
nodeView.RegisterSession(HttpContext.Session);
```

To enable session support in the application, open *Program.cs* and add the following lines:

C# Copy Code

```
builder.Services.AddSession();
...
app.UseSession();
```

5. Initialize the diagram with some OrgChartNodes:

C# Copy Code

```
if (Request.HasFormContentType)
{
    view.LoadFromJson(Request.Form["diagramView1_PostData"]);
    nodeView.LoadFromJson(Request.Form["nodeListView1_PostData"]);
}
else
{
    Diagram diagram = view.Diagram;
    var node1 = new OrgChartNode();
    node1.Bounds = new Rect(25, 15, 60, 25);
    node1.Title = "CEO";
    node1.FullName = "John Smith";
    node1.Details =
        "Our beloved leader.\r\n" +
        "The CEO of this great corporation.";
```

```
    node1.Image = "./images/ceo.png";
    diagram.Nodes.Add(node1);
    nodeView.AddNode(node1);

    var node2 = new OrgChartNode();
    node2.Bounds = new Rect(25, 55, 60, 25);
    node2.Title = "CTO";
    node2.FullName = "Bob Smith";
    node2.Details = "The technology chief of this great corporation.";
    node2.Image = "./images/cto.png";
    diagram.Nodes.Add(node2);
    nodeView.AddNode(node2);

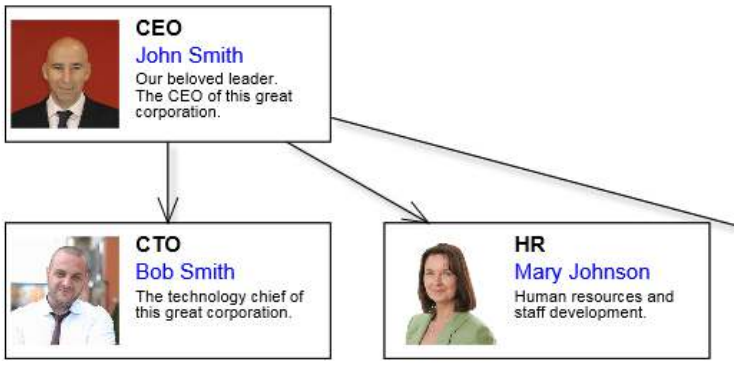
    var node3 = new OrgChartNode();
    node3.Bounds = new Rect(95, 55, 60, 25);
    node3.Title = "HR";
    node3.FullName = "Mary Johnson";
    node3.Details = "Human resources and staff development.";
    node3.Image = "./images/hr.png";
    diagram.Nodes.Add(node3);
    nodeView.AddNode(node3);

    var node4 = new OrgChartNode();
    node4.Bounds = new Rect(175, 55, 60, 25);
    node4.Title = "PR";
    node4.FullName = "Diana Brandson";
    node4.Details = "Public relations and media.";
    node4.Image = "./images/pr.png";
    diagram.Nodes.Add(node4);
    nodeView.AddNode(node4);

    diagram.Factory.CreateDiagramLink(node1, node2);
    diagram.Factory.CreateDiagramLink(node1, node3);
    diagram.Factory.CreateDiagramLink(node3, node4);
}

ViewBag.DiagramView = view;
ViewBag.NodeListView = nodeView;
```

6. If you run the tutorial now, you should see this in your web browser:



Reference

Reference

MindFusion.Animations Namespace

Animation Class

Represents an object that is used to animate diagram items on the canvas.

Syntax

JavaScript — Copy Code

```
// class
Animation.prototype = {}
```

Remarks

For more information on how to use the **Animation** class, please refer to the Animations topic in the **Programming Interface Overview**.

Inheritance Hierarchy

MindFusion.Animations.Animation

See Also

Animation Members
MindFusion.Animations Namespace

Animation Members

Represents an object that is used to animate diagram items on the canvas.

The following tables list the members exposed by the Animation type.

Public Constructors

	Name	Description
	Animation	Initializes a new instance of the **Animation** class.

Public Properties

	Name	Description
	animationType	Gets or sets the type of the animation.
	duration	Gets or sets the duration of the animation function in milliseconds.
	easingType	Gets or sets the easing type of the animation.
	fromValue	Gets or sets the initial value of the property that will be animated.
	isRunning	Indicates whether the animation is currently running.
	param	Gets or sets an optional value passed to the animation function.
	progress	Gets the progress of the current animation iteration.
	repeat	Gets or sets a value indicating whether to restart the animation after it completes.
	reverse	Gets or sets a value indicating whether to swap the from and to values after an animation loop completes.
	toValue	Gets or sets the target value of the property that will be animated.

Public Methods

	Name	Description
	addEventListener	Registers a single event listener on the **Animation**.
	dispose	Sys.IDisposable implementation.
	raiseEvent	Raises an event.
	removeEventListener	Removes a single event listener attached to the **Animation**.
	start	Starts or resumes the animation.
	stop	Stops the animation.

See Also

Animation Class
MindFusion.Animations Namespace

Animation Constructor

Initializes a new instance of the Animation class.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript Copy Code

```
function Animation (item, [options, [onUpdateCallback, [onCustomFunctionCallback]]])
```

Parameters

item
The diagram item to animate.

options
Optional.
An object used to specify various property values for the animation.

onUpdateCallback
Optional.
A callback function called on every animation tick.

onCustomFunctionCallback
Optional.
A callback function called when the animation type is set to Custom.

See Also

Animation Class
Animations
MindFusion.Animations Namespace

Animation Methods

Animation.addEventListener Method

Registers a single event listener on the Animation.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript Copy Code

```
function addEventListener (eventName, handler)
```

Parameters

eventName
The name of the event.

handler
Represents the method that will handle the event specified with eventName.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.dispose Method

Sys.IDisposable implementation.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript Copy Code

```
function dispose ()
```

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.raiseEvent Method

Raises an event.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript Copy Code

```
function raiseEvent (eventName, args)
```

Parameters

eventName
The name of the event to raise.

args
An instance of type EventArgs that holds data for the event specified with eventName.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.removeEventListener Method

Removes a single event listener attached to the Animation.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript Copy Code

```
function removeEventListener (eventName, handler)
```

Parameters

eventName
The name of the event.

handler
Represents the method that handles the event specified with eventName.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.start Method

Starts or resumes the animation.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript Copy Code

```
function start ()
```

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.stop Method

Stops the animation.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript Copy Code

```
function stop ()
```

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation Properties

Animation.animationType Property

Gets or sets the type of the animation.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript — Copy Code

```
get animationType() {}
set animationType(value) {}
```

Property Value

AnimationType

A member of the AnimationType enumeration. The default value is **Linear**.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.duration Property

Gets or sets the duration of the animation function in milliseconds.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript — Copy Code

```
get duration() {}
set duration(value) {}
```

Property Value

Number

The duration of the animation. The default value is 1000.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.easingType Property

Gets or sets the easing type of the animation.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript — Copy Code

```
get easingType() {}
set easingType(value) {}
```

Property Value

EasingType

A member of the EasingType enumeration. The default value is **EaseIn**.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.fromValue Property

Gets or sets the initial value of the property that will be animated.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript Copy Code

```
get fromValue() {}
set fromValue(value) {}
```

Property Value

Object

The initial value.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.isRunning Property

Indicates whether the animation is currently running.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript Copy Code

```
get isRunning() {}
set isRunning(value) {}
```

Property Value

Boolean

true if animation currently in progress, or **false** otherwise.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.param Property

Gets or sets an optional value passed to the animation function.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

```
JavaScript
get param() {}
set param(value) {}
```

Property Value

Object

The optional parameter. The default value is **null**.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.progress Property

Gets the progress of the current animation iteration.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

```
JavaScript
get progress() {}
set progress(value) {}
```

Property Value

Number

Animation progress in 0-1 range.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.repeat Property

Gets or sets a value indicating whether to restart the animation after it completes.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

```
JavaScript
get repeat() {}
set repeat(value) {}
```

Property Value

Boolean

true to loop the animation, or **false** otherwise. The default value is **false**.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.reverse Property

Gets or sets a value indicating whether to swap the from and to values after an animation loop completes.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript Copy Code

```
get reverse() {}
set reverse(value) {}
```

Property Value

Boolean

true to reverse the animation, or **false** otherwise. The default value is **false**.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

Animation.toValue Property

Gets or sets the target value of the property that will be animated.

Namespace: MindFusion.Animations
Assembly: Animation.js

Syntax

JavaScript Copy Code

```
get toValue() {}
set toValue(value) {}
```

Property Value

Object

The target value.

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

AnimationType Enumeration

Specifies the animation function used by an Animation object.

Namespace: MindFusion.Animations
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
AnimationType = {}
```

Members

	Member name	Description
	Linear	Represents an animation that accelerates/deccelerates using a linear function.
	Power	Represents an animation that accelerates/deccelerates through exponentiation formula - x^n where n can be set through the Animation's param property. The default exponent equals 2.
	Exponential	Represents an animation that accelerates/deccelerates using an exponential function. The exponent can be set through the Animation's param property. The default exponent equals 10.
	Circular	Represents an animation that accelerates/deccelerates using a circular function.
	BackBow	Represents an effect that simulates a retraction in the motion of the animation. The amplitude of the retraction can be controlled through the Animation's param property. The default param value is 1.5;
	Bounce	Represents a bouncing effect animation.
	Elastic	Represents an oscillating effect animation. The oscillation frequency can be controlled through the Animation's param property. The default param value is 0.5;
	Custom	Specifies a user-set function. When this value is used the actual animation function must be set through the Animation's custom function callback.

Remarks

Members of this enumeration can be assigned to the animationType property of Animation.

See Also

MindFusion.Animations Namespace

EasingType Enumeration

Specifies the easing applied to an animation.

Namespace: MindFusion.Animations
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
EasingType = {}
```

Members

	Member name	Description
	EaseIn	Follows the animation function associated with the Animation.
	EaseOut	Inverses the animation function associated with the **Animation**.
	EaseInOut	Uses EaseIn for the first half the the Animation's duration and EaseOut for the second half.

	EaseOutIn	Uses EaseOut for the first half the the Animation's duration and EaseIn for the second half.

Remarks

Members of this enumeration can be assigned to the easingType property of **Animation**.

See Also

MindFusion.Animations Namespace

Events Class

Defines all events raised in the Animations namespace.

Syntax

JavaScript Copy Code

```
// class
Events.prototype = {}
```

Inheritance Hierarchy

MindFusion.Animations.Events

See Also

Events Members
MindFusion.Animations Namespace

Events Members

Defines all events raised in the Animations namespace.

The following tables list the members exposed by the Events type.

Public Properties

	Name	Description
	animationComplete	Raised when an Animation is complete.

See Also

Events Class
MindFusion.Animations Namespace

Events Fields

Events Properties

Events.animationComplete Property

Raised when an Animation is complete.

Namespace: MindFusion.Animations
Assembly: Events.js

Syntax

```
JavaScript                                                    Copy Code

get animationComplete() {}
set animationComplete(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Animations Namespace

MindFusion.Collections Namespace

Grid Class

Represents a collection of tabular data.

Syntax

```
JavaScript                                                    Copy Code

// class
Grid.prototype = {}
```

Inheritance Hierarchy

See Also

Grid Members
MindFusion.Collections Namespace

Grid Members

Represents a collection of tabular data.

The following tables list the members exposed by the Grid type.

Public Constructors

	Name	Description
	Grid	Initializes a new instance of the Grid class.

See Also

Grid Class
MindFusion.Collections Namespace

Grid Constructor

Initializes a new instance of the Grid class.

Namespace: MindFusion.Collections
Assembly: Grid.js

Syntax

JavaScript — Copy Code

```
function Grid ()
```

See Also

Grid Class
MindFusion.Collections Namespace

Queue Class

Represents a first-in, first-out collection of objects.

Syntax

JavaScript — Copy Code

```
// class
Queue.prototype = {}
```

Inheritance Hierarchy

See Also

Queue Members
MindFusion.Collections Namespace

Queue Members

Represents a first-in, first-out collection of objects.

The following tables list the members exposed by the Queue type.

Public Constructors

	Name	Description
	Queue	Initializes a new instance of the **Queue** class.

Public Methods

	Name	Description
	dequeue	Removes and returns the object at the beginning of the queue.
	enqueue	Adds an object to the end of the queue.
	getSize	Gets the number of elements in the queue.

See Also

Queue Class
MindFusion.Collections Namespace

Queue Constructor

Initializes a new instance of the Queue class.

Namespace: MindFusion.Collections
Assembly: Queue.js

Syntax

JavaScript Copy Code

```
function Queue ()
```

See Also

Queue Class
MindFusion.Collections Namespace

Queue Methods

Queue.dequeue Method

Removes and returns the object at the beginning of the queue.

Namespace: MindFusion.Collections
Assembly: Queue.js

Syntax

JavaScript Copy Code

```
function dequeue ()
```

Return Value

The object that is removed from the beginning of the queue.

Remarks

If the queue is empty this method will throw an exception.

See Also

Queue Members
Queue Class
MindFusion.Collections Namespace

Queue.enqueue Method

Adds an object to the end of the queue.

Namespace: MindFusion.Collections
Assembly: Queue.js

Syntax

JavaScript Copy Code

```
function enqueue (value)
```

Parameters

value
The object to add.

See Also

Queue Members
Queue Class
MindFusion.Collections Namespace

Queue.getSize Method

Gets the number of elements in the queue.

Namespace: MindFusion.Collections
Assembly: Queue.js

Syntax

JavaScript Copy Code

```
function getSize ()
```

Return Value

See Also

Queue Members
Queue Class
MindFusion.Collections Namespace

MindFusion.Controls Namespace

Alignment Enumeration

Specifies the alignment of text relative to its layout rectangle.

Namespace: MindFusion.Controls
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
Alignment = {}
```

Members

	Member name	Description
	Near	The text is drawn in the near corner of the layout rectangle.
	Center	The text is drawn in the center of the layout rectangle.
	Far	The text is drawn in the far corner of the layout rectangle.

See Also

MindFusion.Controls Namespace

CancelEventArgs Class

Provides a value to use with cancellable events.

Syntax

JavaScript Copy Code

```
// class
CancelEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.EventArgs
  MindFusion.Controls.CancelEventArgs
    MindFusion.Diagramming.ItemEventArgs
    MindFusion.Diagramming.SelectionEventArgs
    MindFusion.Diagramming.TreeItemEventArgs

See Also

CancelEventArgs Members
MindFusion.Controls Namespace

CancelEventArgs Members

Provides a value to use with cancellable events.

The following tables list the members exposed by the CancelEventArgs type.

Public Constructors

	Name	Description
	CancelEventArgs	Initializes a new instance of the CancelEventArgs class.

Public Properties

	Name	Description
	cancel	Gets or sets a value indicating whether to allow the current operation.
	handled	Gets or sets a value indicating whether the event has been handled. (Inherited from EventArgs.)

See Also

CancelEventArgs Class
MindFusion.Controls Namespace

CancelEventArgs Constructor

Initializes a new instance of the CancelEventArgs class.

Namespace: MindFusion.Controls
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function CancelEventArgs ()
```

See Also

CancelEventArgs Class
MindFusion.Controls Namespace

CancelEventArgs Properties

CancelEventArgs.cancel Property

Gets or sets a value indicating whether to allow the current operation.

Namespace: MindFusion.Controls
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get cancel() {}
set cancel(value) {}
```

Property Value

Boolean

true to cancel the operation, or **false** otherwise. The default value is **false**.

See Also

CancelEventArgs Members
CancelEventArgs Class
MindFusion.Controls Namespace

Canvas Class

Handles drawing on a CanvasRenderingContext2D.

Syntax

JavaScript Copy Code

```
// class
Canvas.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.Disposable
  MindFusion.Controls.Canvas

See Also

Canvas Members
MindFusion.Controls Namespace

Canvas Members

Handles drawing on a CanvasRenderingContext2D.

The following tables list the members exposed by the Canvas type.

Public Constructors

	Name	Description
	Canvas	Initializes a new instance of the Canvas class.

Public Properties

	Name	Description

	bounds	Gets the underlying **Canvas** element's logical bounds.
	element	Returns a reference to the control's DOM element. (Inherited from Disposable.)
	enabled	Gets or sets whether mouse events are enabled. (Inherited from Disposable.)
	measureUnit	Gets or sets the unit of measure used for logical coordinates.
	minVisibleFontSize	Gets or sets a threshold value that hides text if scaled font sizes become smaller.
	scale	Gets the current scale of this **Canvas**.

Public Methods

	Name	Description
	addEventListener	Registers a single event listener on the instance. (Inherited from Disposable.)
	dispose	Overrides Disposable.dispose
	invalidate	Invalidates the canvas or a region of the canvas, causing it to be repainted.
	raiseEvent	Raises an event. (Inherited from Disposable.)
	removeEventListener	Removes a single event listener attached to the instance. (Inherited from Disposable.)
	repaint	Repaints the canvas.

See Also

Canvas Class
MindFusion.Controls Namespace

Canvas Constructor

Initializes a new instance of the Canvas class.

Namespace: MindFusion.Controls
Assembly: Canvas.js

Syntax

JavaScript Copy Code

```
function Canvas ()
```

See Also

Canvas Class
MindFusion.Controls Namespace

Canvas Methods

Canvas.dispose Method

Overrides Disposable.dispose

Namespace: MindFusion.Controls
Assembly: Canvas.js

Syntax

JavaScript Copy Code

```
function dispose ()
```

See Also

Canvas Members
Canvas Class
MindFusion.Controls Namespace

Canvas.invalidate Method

Invalidates the canvas or a region of the canvas, causing it to be repainted.

Namespace: MindFusion.Controls
Assembly: Canvas.js

Syntax

JavaScript Copy Code

```
function invalidate ([rect, [force]])
```

Parameters

rect
Optional.
Rect. A Rect instance specifying the region that should be repainted. If not specified, the whole diagram will be repainted.

force
Optional.
Boolean. **true** to force redraw even if currently painting, or **false** otherwise.

See Also

Canvas Members
Canvas Class
MindFusion.Controls Namespace

Canvas.repaint Method

Repaints the canvas.

Namespace: MindFusion.Controls
Assembly: Canvas.js

Syntax

JavaScript Copy Code

```
function repaint ([printOptions])
```

Parameters

printOptions
Optional.
Object. For internal use.

See Also

Canvas Members
Canvas Class
MindFusion.Controls Namespace

Canvas Properties

Canvas.bounds Property

Gets the underlying Canvas element's logical bounds.

Namespace: MindFusion.Controls
Assembly: Canvas.js

Syntax

JavaScript — Copy Code

```
get bounds() {}
set bounds(value) {}
```

Property Value

Rect

A Rect instance specifying canvas bounds in current measureUnit.

See Also

Canvas Members
Canvas Class
MindFusion.Controls Namespace
Diagram.enableSpatialIndex Property

Canvas.measureUnit Property

Gets or sets the unit of measure used for logical coordinates.

Namespace: MindFusion.Controls
Assembly: Canvas.js

Syntax

JavaScript — Copy Code

```
get measureUnit() {}
set measureUnit(value) {}
```

Property Value

GraphicsUnit

A member of the GraphicsUnit enumeration.

See Also

Canvas Members
Canvas Class
MindFusion.Controls Namespace

Canvas.minVisibleFontSize Property

Gets or sets a threshold value that hides text if scaled font sizes become smaller.

Namespace: MindFusion.Controls
Assembly: Canvas.js

Syntax

JavaScript — Copy Code

```
get minVisibleFontSize() {}
set minVisibleFontSize(value) {}
```

Property Value

Number

A value specifying the smallest visible font size.

See Also

Canvas Members
Canvas Class
MindFusion.Controls Namespace

Canvas.scale Property

Gets the current scale of this Canvas.

Namespace: MindFusion.Controls
Assembly: Canvas.js

Syntax

JavaScript Copy Code

```
get scale() {}
set scale(value) {}
```

Property Value

Number

The canvas scale.

See Also

Canvas Members
Canvas Class
MindFusion.Controls Namespace

CanvasControl Class

The CanvasControl class represents a wrapper class for the HTML5 Canvas element.

Syntax

JavaScript Copy Code

```
// class
CanvasControl.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.Disposable
  MindFusion.Controls.Control
    MindFusion.Controls.CanvasControl
      MindFusion.Controls.ZoomControl
      MindFusion.Diagramming.ListViewBase
      MindFusion.Diagramming.Overview

See Also

CanvasControl Members
MindFusion.Controls Namespace

CanvasControl Members

The CanvasControl class represents a wrapper class for the HTML5 Canvas element.

The following tables list the members exposed by the CanvasControl type.

Public Constructors

	Name	Description
	CanvasControl	Initializes a new instance of the CanvasControl class.

Public Properties

	Name	Description
	element	Returns a reference to the control's DOM element. (Inherited from Control.)
	enabled	Gets or sets whether mouse events are enabled. (Inherited from Control.)
	licenseLocation	Gets or sets the location of the license key file. (Inherited from Control.)
	measureUnit	Gets the unit of measure used for logical coordinates.

Public Methods

	Name	Description
	addEventListener	Registers a single event listener on the instance. (Inherited from Disposable.)
S	addHandlers	Adds an event listener to a DOM element. (Inherited from Control.)
S	clearHandlers	Removes all event listeners from a DOM element. (Inherited from Control.)
	clientToDoc	Transforms a point from client to document coordinates.
	docToClient	Transforms a point from document to client coordinates.
S	fromJson	Constructs a JavaScript object from a JSON string. (Inherited from Control.)
	raiseEvent	Raises an event. (Inherited from Disposable.)
	registerForSubmit	Registers an onsubmit handler for the Control's parent html form to flush postback data. (Inherited from Control.)
	removeEventListener	Removes a single event listener attached to the instance. (Inherited from Disposable.)
S	toJson	Converts a JavaScript object to a JSON string. (Inherited from Control.)

Public Events

	Name	Description
	controlLoaded	Raised when the control is loaded. (Inherited from Control.)
	controlUnloaded	Raised when the control is unloaded. (Inherited from Control.)
	controlUnloading	Raised when the control is unloading. (Inherited from Control.)

See Also

CanvasControl Class
MindFusion.Controls Namespace

CanvasControl Constructor

Initializes a new instance of the CanvasControl class.

Namespace: MindFusion.Controls
Assembly: CanvasControl.js

Syntax

JavaScript Copy Code

```
function CanvasControl (element, [canvas])
```

Parameters

element
HTMLCanvasElement. The Canvas DOM Element this CanvasControl is associated with.

canvas
Optional.
Canvas. The Canvas instance this CanvasControl is associated with.

See Also

CanvasControl Class
MindFusion.Controls Namespace

CanvasControl Methods

CanvasControl.clientToDoc Method

Transforms a point from client to document coordinates.

Namespace: MindFusion.Controls
Assembly: CanvasControl.js

Syntax

JavaScript Copy Code

```
function clientToDoc (point)
```

Parameters

point
Point. The point to transform.

Return Value

Point. The transformed point.

See Also

CanvasControl Members
CanvasControl Class
MindFusion.Controls Namespace

CanvasControl.docToClient Method

Transforms a point from document to client coordinates.

Namespace: MindFusion.Controls
Assembly: CanvasControl.js

Syntax

JavaScript Copy Code

```
function docToClient (point)
```

Parameters

point
Point. The point to transform.

Return Value

Point. The transformed point.

See Also

CanvasControl Members
CanvasControl Class
MindFusion.Controls Namespace

CanvasControl Properties

CanvasControl.measureUnit Property

Gets the unit of measure used for logical coordinates.

Namespace: MindFusion.Controls
Assembly: CanvasControl.js

Syntax

JavaScript Copy Code

```
get measureUnit() {}
set measureUnit(value) {}
```

Property Value

GraphicsUnit

An instance of the GraphicsUnit class.

See Also

CanvasControl Members
CanvasControl Class
MindFusion.Controls Namespace

CanvasControl Events

Control.controlLoaded Event

Raised when the control is loaded.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
diagramView.controlLoaded.addEventListener(onViewLoaded);

function onViewLoaded(sender, args)
{
}
```

See Also

Control Members
Control Class
MindFusion.Controls Namespace

Control.controlUnloaded Event

Raised when the control is unloaded.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
diagramView.controlUnloaded.addEventListener(onViewUnloaded);

function onViewUnloaded(sender, args)
{
}
```

Remarks

This event is raised by the dispose method. In vanilla component, that usually happens when the page unloads. In diagram wrapper components for JavaScrript UI frameworks, it's raised by component's lifecycle hooks indicating that DOM element is being detached, such as **ngOnDestroy**.

See Also

Control Members
Control Class
MindFusion.Controls Namespace

Control.controlUnloading Event

Raised when the control is unloading.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
diagramView.controlUnloading.addEventListener(onViewUnloading);

function onViewUnloading(sender, args)
{
}
```

Remarks

This event is raised by the dispose method, before detaching DOM elements. In vanilla component, that usually happens when the page unloads. In diagram wrapper components for JavaScrript UI frameworks, it's raised by component's lifecycle hooks indicating that DOM element is being detached, such as **ngOnDestroy**.

See Also

Control Members
Control Class
MindFusion.Controls Namespace

Control Class

A base class for MindFusion controls.

Syntax

JavaScript Copy Code

```
// class
Control.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.Disposable
  MindFusion.Controls.Control
    MindFusion.Controls.CanvasControl
    MindFusion.Diagramming.Ruler

See Also

Control Members
MindFusion.Controls Namespace

Control Members

A base class for MindFusion controls.

The following tables list the members exposed by the Control type.

Public Constructors

	Name	Description
	Control	Initializes a new instance of the Control class.

Public Properties

	Name	Description
	element	Returns a reference to the control's DOM element.
	enabled	Gets or sets whether mouse events are enabled.
	licenseLocation	Gets or sets the location of the license key file.

Public Methods

	Name	Description
	addEventListener	Registers a single event listener on the instance. (Inherited from Disposable.)
S	addHandlers	Adds an event listener to a DOM element.
S	clearHandlers	Removes all event listeners from a DOM element.
S	fromJson	Constructs a JavaScript object from a JSON string.
	raiseEvent	Raises an event. (Inherited from Disposable.)
	registerForSubmit	Registers an onsubmit handler for the Control's parent html form to flush postback data.
	removeEventListener	Removes a single event listener attached to the instance. (Inherited from Disposable.)
S	toJson	Converts a JavaScript object to a JSON string.

Public Events

	Name	Description
	controlLoaded	Raised when the control is loaded.
	controlUnloaded	Raised when the control is unloaded.
	controlUnloading	Raised when the control is unloading.

See Also

Control Class
MindFusion.Controls Namespace

Control Constructor

Initializes a new instance of the Control class.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
function Control (element)
```

Parameters

element
HTMLElement. The DOM Element this Control is associated with.

See Also

Control Class
MindFusion.Controls Namespace

Control Methods

Control.addHandlers Method

Adds an event listener to a DOM element.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
function addHandlers ()
```

See Also

Control Members
Control Class
MindFusion.Controls Namespace

Control.clearHandlers Method

Removes all event listeners from a DOM element.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

```
JavaScript
function clearHandlers ()
```

See Also

Control Members
Control Class
MindFusion.Controls Namespace

Control.fromJson Method

Constructs a JavaScript object from a JSON string.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

```
JavaScript
function fromJson ()
```

Return Value

Object. object The corresponding to the given JSON text.

See Also

Control Members
Control Class
MindFusion.Controls Namespace

Control.registerForSubmit Method

Registers an onsubmit handler for the Control's parent html form to flush postback data.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

```
JavaScript
function registerForSubmit (id)
```

Parameters

id
String. The id of the hidden field to flush the data to.

See Also

Control Members
Control Class
MindFusion.Controls Namespace

Control.toJson Method

Converts a JavaScript object to a JSON string.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
function toJson (object)
```

Parameters

object
Object. The object to stringify.

Return Value

String. The JSON string.

See Also

Control Members
Control Class
MindFusion.Controls Namespace

Control Properties

Control.element Property

Returns a reference to the control's DOM element.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
get element() {}
set element(value) {}
```

Property Value

HTMLElement

The HTML element wrapped by this control.

See Also

Control Members
Control Class
MindFusion.Controls Namespace

Control.enabled Property

Gets or sets whether mouse events are enabled.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
get enabled() {}
set enabled(value) {}
```

Property Value

Boolean

true if the control should respond to mouse events, or false otherwise.

See Also

Control Members
Control Class
MindFusion.Controls Namespace

Control.licenseLocation Property

Gets or sets the location of the license key file.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
get licenseLocation() {}
set licenseLocation(value) {}
```

Property Value

String

the location of license key file.

Remarks

Control.licenseLocation can also be set as a static property, which applies the key to all instances of controls from MindFusion.JavaScript Pack.

See Also

Control Members
Control Class
MindFusion.Controls Namespace

Control Events

Disposable Class

Disposable abstract class.

Syntax

JavaScript Copy Code

```
// class
Disposable.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.Disposable
MindFusion.Controls.Canvas
MindFusion.Controls.Control

See Also

Disposable Members
MindFusion.Controls Namespace

Disposable Members

Disposable abstract class.

The following tables list the members exposed by the Disposable type.

Public Properties

	Name	Description
	element	Returns a reference to the control's DOM element.
	enabled	Gets or sets whether mouse events are enabled.

Public Methods

	Name	Description
	addEventListener	Registers a single event listener on the instance.
	raiseEvent	Raises an event.
	removeEventListener	Removes a single event listener attached to the instance.

See Also

Disposable Class
MindFusion.Controls Namespace

Disposable Methods

Disposable.addEventListener Method

Registers a single event listener on the instance.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
function addEventListener (eventName, handler, [element])
```

Parameters

eventName
String. The name of the event.

handler
Method. Represents the method that will handle the event specified with eventName.

element
Optional.
Object. For internal use.

See Also

Disposable Members
Disposable Class
MindFusion.Controls Namespace
removeEventListener Method

Disposable.raiseEvent Method

Raises an event.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript — Copy Code

```
function raiseEvent (eventName, args, [element])
```

Parameters

eventName
String. The name of the event to raise.

args
EventArgs. An instance of type EventArgs that holds data for the event specified with eventName.

element
Optional.
Object. For internal use.

See Also

Disposable Members
Disposable Class
MindFusion.Controls Namespace

Disposable.removeEventListener Method

Removes a single event listener attached to the instance.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript — Copy Code

```
function removeEventListener (eventName, handler, [element])
```

Parameters

eventName
String. The name of the event.

handler
Method. Represents the method that handles the event specified with eventName.

element
Optional.
Object. For internal use.

See Also

Disposable Members
Disposable Class
MindFusion.Controls Namespace
addEventListener Method

Disposable Properties

Disposable.element Property

Returns a reference to the control's DOM element.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
get element() {}
set element(value) {}
```

Property Value

HTMLElement

The HTML element wrapped by this control.

See Also

Disposable Members
Disposable Class
MindFusion.Controls Namespace

Disposable.enabled Property

Gets or sets whether mouse events are enabled.

Namespace: MindFusion.Controls
Assembly: Control.js

Syntax

JavaScript Copy Code

```
get enabled() {}
set enabled(value) {}
```

Property Value

Boolean

true if the control should respond to mouse events, or false otherwise.

See Also

Disposable Members
Disposable Class
MindFusion.Controls Namespace

DomUtils Class

Contains DOM-related helper functions.

Syntax

JavaScript Copy Code

```
// class
DomUtils.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.DomUtils

See Also

DomUtils Members
MindFusion.Controls Namespace

DomUtils Members

Contains DOM-related helper functions.

The following tables list the members exposed by the DomUtils type.

Public Methods

	Name	Description
	formatString	Formats the specified string according to the specified parameters.
	getBounds	Gets the bounds of the specified element.
	getElementAttributes	Gets an object containing element attributes that have values.
	HTMLtoXHTML	Converts HTML to XHMTL.
	loadImageList	Loads the images from the specified list of image url-s.
	setBounds	Sets the bounds of the specified element.
	setSize	Sets the size of the specified element.
	toDataUrl	Converts an image to base64 string.

See Also

DomUtils Class
MindFusion.Controls Namespace

DomUtils Methods

DomUtils.formatString Method

Formats the specified string according to the specified parameters.

Namespace: MindFusion.Controls
Assembly: DomUtils.js

Syntax

JavaScript Copy Code

```
function formatString ()
```

See Also

DomUtils Members
DomUtils Class
MindFusion.Controls Namespace

DomUtils.getBounds Method

Gets the bounds of the specified element.

Namespace: MindFusion.Controls
Assembly: DomUtils.js

Syntax

JavaScript Copy Code

```
function getBounds (element, [parent, [includeScroll]])
```

Parameters

element
HTMLElement. The element to check.

parent
Optional.
HTMLElement. The parent of the element. If supplied, the returned bounds will be relative to the parent element bounds.

includeScroll
Optional.
Boolean. For internal use.

Return Value

Rect. The bounding rectangle.

See Also

DomUtils Members
DomUtils Class
MindFusion.Controls Namespace

DomUtils.getElementAttributes Method

Gets an object containing element attributes that have values.

Namespace: MindFusion.Controls
Assembly: DomUtils.js

Syntax

JavaScript Copy Code

```
function getElementAttributes ()
```

See Also

DomUtils Members
DomUtils Class
MindFusion.Controls Namespace

DomUtils.HTMLtoXHTML Method

Converts HTML to XHMTL.

Namespace: MindFusion.Controls
Assembly: DomUtils.js

Syntax

JavaScript Copy Code

```
function HTMLtoXHTML (html)
```

Parameters

html
String. The HTML to convert.

Return Value

String. The resulting XHTML.

See Also

DomUtils Members
DomUtils Class
MindFusion.Controls Namespace

DomUtils.loadImageList Method

Loads the images from the specified list of image url-s.

Namespace: MindFusion.Controls
Assembly: DomUtils.js

Syntax

JavaScript Copy Code

```
function loadImageList (images)
```

Parameters

images
Array. An array of image url-s.

Return Value

Promise. Promise object contaning an array of images in base64 format.

See Also

DomUtils Members
DomUtils Class
MindFusion.Controls Namespace

DomUtils.setBounds Method

Sets the bounds of the specified element.

Namespace: MindFusion.Controls
Assembly: DomUtils.js

Syntax

JavaScript Copy Code

```
function setBounds (element, bounds)
```

Parameters

element
HTMLElement. The element.

bounds
Rect. The bounding rectangle.

See Also

DomUtils Members
DomUtils Class
MindFusion.Controls Namespace

DomUtils.setSize Method

Sets the size of the specified element.

Namespace: MindFusion.Controls
Assembly: DomUtils.js

Syntax

JavaScript Copy Code

```
function setSize (element, size)
```

Parameters

element
HTMLElement. The element.

size
Size. The new size.

See Also

DomUtils Members
DomUtils Class
MindFusion.Controls Namespace

DomUtils.toDataUrl Method

Converts an image to base64 string.

Namespace: MindFusion.Controls
Assembly: DomUtils.js

Syntax

JavaScript Copy Code

```
function toDataUrl (url, callback)
```

Parameters

url
String. The url of the image.

callback
function. The callback function.

Return Value

String. The base64-encoded string, representing the image.

See Also

DomUtils Members
DomUtils Class
MindFusion.Controls Namespace

EventArgs Class

The base type of classes that define arguments passed to event handler functions.

Syntax

JavaScript Copy Code

```
// class
EventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.EventArgs
MindFusion.Controls.CancelEventArgs
MindFusion.Diagramming.DiagramEventArgs
MindFusion.Diagramming.HeaderEventArgs

MindFusion.Diagramming.InplaceEditEventArgs
MindFusion.Diagramming.SelectionChangedEventArgs

See Also

EventArgs Members
MindFusion.Controls Namespace

EventArgs Members

The base type of classes that define arguments passed to event handler functions.

The following tables list the members exposed by the EventArgs type.

Public Constructors

	Name	Description
	EventArgs	Initializes a new instance of the EventArgs class.

Public Properties

	Name	Description
	handled	Gets or sets a value indicating whether the event has been handled.

See Also

EventArgs Class
MindFusion.Controls Namespace

EventArgs Constructor

Initializes a new instance of the EventArgs class.

Namespace: MindFusion.Controls
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function EventArgs ()
```

See Also

EventArgs Class
MindFusion.Controls Namespace

EventArgs Properties

EventArgs.handled Property

Gets or sets a value indicating whether the event has been handled.

Namespace: MindFusion.Controls
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get handled() {}
set handled(value) {}
```

Property Value

Boolean

true to indicate that the event has been handled, or false otherwise.

See Also

EventArgs Members
EventArgs Class
MindFusion.Controls Namespace

EventDispatcher Class

Represents a dispatcher for an event.

Syntax

JavaScript Copy Code

```
// class
EventDispatcher.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.EventDispatcher

See Also

EventDispatcher Members
MindFusion.Controls Namespace

EventDispatcher Members

Represents a dispatcher for an event.

The following tables list the members exposed by the EventDispatcher type.

Public Constructors

	Name	Description
	EventDispatcher	Creates a new instance of the EventDispatcher class.

Public Methods

	Name	Description
	addEventListener	Subcribes an event listener to this event.
	raiseEvent	Raises the event.
	raiseEventAsync	Raises the event asynchronously.
	removeEventListener	Removes an event listener from this event.

See Also

EventDispatcher Class
MindFusion.Controls Namespace

EventDispatcher Constructor

Creates a new instance of the EventDispatcher class.

Namespace: MindFusion.Controls
Assembly: EventDispatcher.js

Syntax

JavaScript Copy Code

```
function EventDispatcher ()
```

See Also

EventDispatcher Class
MindFusion.Controls Namespace

EventDispatcher Methods

EventDispatcher.addEventListener Method

Subcribes an event listener to this event.

Namespace: MindFusion.Controls
Assembly: EventDispatcher.js

Syntax

JavaScript Copy Code

```
function addEventListener ()
```

See Also

EventDispatcher Members
EventDispatcher Class
MindFusion.Controls Namespace

EventDispatcher.raiseEvent Method

Raises the event.

Namespace: MindFusion.Controls
Assembly: EventDispatcher.js

Syntax

JavaScript Copy Code

```
function raiseEvent ()
```

See Also

EventDispatcher Members
EventDispatcher Class
MindFusion.Controls Namespace

EventDispatcher.raiseEventAsync Method

Raises the event asynchronously.

Namespace: MindFusion.Controls
Assembly: EventDispatcher.js

Syntax

JavaScript Copy Code

```
function raiseEventAsync ()
```

See Also

EventDispatcher Members
EventDispatcher Class
MindFusion.Controls Namespace

EventDispatcher.removeEventListener Method

Removes an event listener from this event.

Namespace: MindFusion.Controls
Assembly: EventDispatcher.js

Syntax

JavaScript Copy Code

```
function removeEventListener ()
```

See Also

EventDispatcher Members
EventDispatcher Class
MindFusion.Controls Namespace

Events Class

Defines all events raised in the Controls namespace.

Syntax

JavaScript Copy Code

```
// class
Events.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.Events

See Also

Events Members
MindFusion.Controls Namespace

Events Members

Defines all events raised in the Controls namespace.

The following tables list the members exposed by the Events type.

Public Properties

	Name	Description

	controlLoaded	Raised when the control is loaded.
	controlUnloaded	Raised when the control is unloaded.
	controlUnloading	Raised before the control is unloaded.

See Also

Events Class
MindFusion.Controls Namespace

Events Properties

Events.controlLoaded Property

Raised when the control is loaded.

Namespace: MindFusion.Controls
Assembly: EventDispatcher.js

Syntax

JavaScript Copy Code

```
get controlLoaded() {}
set controlLoaded(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Controls Namespace

Events.controlUnloaded Property

Raised when the control is unloaded.

Namespace: MindFusion.Controls
Assembly: EventDispatcher.js

Syntax

JavaScript Copy Code

```
get controlUnloaded() {}
set controlUnloaded(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Controls Namespace

Events.controlUnloading Property

Raised before the control is unloaded.

Namespace: MindFusion.Controls
Assembly: EventDispatcher.js

Syntax

JavaScript — Copy Code

```
get controlUnloading() {}
set controlUnloading(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Controls Namespace

MouseButton Enumeration

Specifies which button was pressed to trigger the event.

Namespace: MindFusion.Controls
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
MouseButton = {}
```

Members

	Member name	Description
	Main	Main button pressed, usually the left button or the un-initialized state.
	Auxiliary	Auxiliary button pressed, usually the wheel button or the middle button.
	Secondary	Secondary button pressed, usually the right button.
	Fourth	Fourth button, typically the Browser Back button.
	Fifth	Fifth button, typically the Browser Forward button.

See Also

MindFusion.Controls Namespace

MouseCursors Enumeration

Specifies the type of the mouse cursor.

Namespace: MindFusion.Controls
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
MouseCursors = {}
```

See Also

MindFusion.Controls Namespace

TickPosition Enumeration

Specifies the position of trackbar ticks.

Namespace: MindFusion.Controls
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
TickPosition = {}
```

Members

	Member name	Description
	None	There are no ticks rendered.
	Left	Ticks are rendered on the left side of the trackbar.
	Right	Ticks are rendered on the right side of the trackbar.
	Both	Ticks are rendered on both sides of the trackbar.

Remarks

Members of this enumeration can be assigned to the tickPosition property of ZoomControl.

See Also

MindFusion.Controls Namespace

WebComponents Class

Implements registration of web components.

Syntax

JavaScript Copy Code

```
// class
WebComponents.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.WebComponents

See Also

WebComponents Members
MindFusion.Controls Namespace

WebComponents Members

Implements registration of web components.

The following tables list the members exposed by the WebComponents type.

Public Methods

	Name	Description
	register	Registers the controls in the package as web components.

See Also

WebComponents Class
MindFusion.Controls Namespace

WebComponents Methods

WebComponents.register Method

Registers the controls in the package as web components.

Namespace: MindFusion.Controls
Assembly: WebComponents.js

Syntax

JavaScript Copy Code

```
function register ()
```

See Also

WebComponents Members
WebComponents Class
MindFusion.Controls Namespace

ZoomControl Class

The **ZoomControl** lets users zoom and pan a target view control interactively.

Syntax

JavaScript Copy Code

```
// class
ZoomControl.prototype = {}
```

Inheritance Hierarchy

MindFusion.Dom.Control
  MindFusion.Controls.ZoomControl

See Also

ZoomControl Members
MindFusion.Controls Namespace

ZoomControl Members

The ZoomControl lets users zoom and pan a target view control interactively.

The following tables list the members exposed by the **ZoomControl** type.

Public Constructors

	Name	Description
	ZoomControl	Creates an instance of **ZoomControl**.

Public Properties

	Name	Description
	activeColor	Gets or sets the color used to render depressed buttons.
	autoPostBack	Gets or sets a value indicating whether the control will post back to the server when the control's value has changed.
	backColor	Gets or sets the background color of the control.
	borderColor	Gets or sets the color of **ZoomControl** elements' borders.
	cornerRadius	Gets or sets the corner radius of rounded child elements.
	element	Returns a reference to the control's DOM element. (Inherited from Control.)
	enabled	Gets or sets whether mouse events are enabled.
	fill	Gets or sets the color used to fill the **ZoomControl** elements.
	innerColor	Gets or sets the color of plus, minus and arrow icons.
	licenseLocation	Gets or sets the location of the license key file. (Inherited from Control.)
	maxZoomFactor	Gets or sets the maximum zoom level allowed to set through this control.
	measureUnit	Gets the unit of measure used for logical coordinates. (Inherited from CanvasControl.)
	minZoomFactor	Gets or sets the minimum zoom level allowed to set through this control.
	padding	Gets or sets the padding of the control's contents.
	scrollStep	Gets or sets the scroll offset added when users click the pan arrows.
	shadowColor	Gets or sets the color of the control elements' shadow.
	showLabel	Gets or sets a value indicating whether the label that shows the current zoom level should be visible.
	snapToZoomStep	Gets or sets a value indicating whether the trackbar should snap to zoomStep values when dragged.
	target	Gets or sets the control modified by this **ZoomControl**.
	textColor	Gets or sets the color of the the label that shows the current zoom level.
	tickPosition	Gets or sets the current tick position of the trackbar.
	zoomFactor	Gets or sets the zoom factor.
	zoomStep	Gets or sets the amount by which to change zoom level when + and - buttons are clicked.

Public Methods

	Name	Description

	addEventListener	Registers a single event listener on the instance. (Inherited from Disposable.)
S	addHandlers	Adds an event listener to a DOM element. (Inherited from Control.)
S	clearHandlers	Removes all event listeners from a DOM element. (Inherited from Control.)
	clientToDoc	Transforms a point from client to document coordinates. (Inherited from CanvasControl.)
S	create	Creates and initializes a new **ZoomControl** from the specified element. This method is static and can be called without creating an instance of the class.
	docToClient	Transforms a point from document to client coordinates. (Inherited from CanvasControl.)
S	find	Returns the specified **ZoomControl** object. This member is static and can be invoked without creating an instance of the class.
S	fromJson	Constructs a JavaScript object from a JSON string. (Inherited from Control.)
	raiseEvent	Raises an event. (Inherited from Disposable.)
	registerForSubmit	Registers an onsubmit handler for the Control's parent html form to flush postback data. (Inherited from Control.)
	removeEventListener	Removes a single event listener attached to the instance. (Inherited from Disposable.)
S	toJson	Converts a JavaScript object to a JSON string. (Inherited from Control.)

Public Events

	Name	Description
	controlLoaded	Raised when the control is loaded. (Inherited from Control.)
	controlUnloaded	Raised when the control is unloaded. (Inherited from Control.)
	controlUnloading	Raised when the control is unloading. (Inherited from Control.)

See Also

ZoomControl Class
MindFusion.Controls Namespace

ZoomControl Constructor

Creates an instance of ZoomControl.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
function ZoomControl (element)
```

Parameters

element
HTMLCanvasElement. The Canvas DOM Element this ZoomControl is associated with.

See Also

ZoomControl Methods

ZoomControl.create Method

Creates and initializes a new ZoomControl from the specified element. This method is static and can be called without creating an instance of the class.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
function create (element)
```

Parameters

element
HTMLCanvasElement. The DOM element that the zoomControl should be attached to.

Return Value

A ZoomControl object that represents the newly created zoom control.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.find Method

Returns the specified ZoomControl object. This member is static and can be invoked without creating an instance of the class.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
function find (id)
```

Parameters

id
String. The id of the ZoomControl's DOM element.

Return Value

A **ZoomControl** object with the specified id, if found, or **null** otherwise.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl Properties

ZoomControl.activeColor Property

Gets or sets the color used to render depressed buttons.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get activeColor() {}
set activeColor(value) {}
```

Property Value

String

The color used to render depressed buttons.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.autoPostBack Property

Gets or sets a value indicating whether the control will post back to the server when the control's value has changed.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get autoPostBack() {}
set autoPostBack(value) {}
```

Property Value

Boolean

true if the control should post back, or **false** otherwise.

Remarks

Used to implement auto-post back mode in MindFusion ASP.NET controls.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.backColor Property

Gets or sets the background color of the control.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get backColor() {}
set backColor(value) {}
```

Property Value

String

The color of the control background.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.borderColor Property

Gets or sets the color of ZoomControl elements' borders.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get borderColor() {}
set borderColor(value) {}
```

Property Value

String

The color of ZoomControl elements' borders.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.cornerRadius Property

Gets or sets the corner radius of rounded child elements.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get cornerRadius() {}
set cornerRadius(value) {}
```

Property Value

Number

The corner radius of rounded child elements.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.enabled Property

Gets or sets whether mouse events are enabled.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get enabled() {}
set enabled(value) {}
```

Property Value

Boolean

true to respond to mouse events, or **false** otherwise.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.fill Property

Gets or sets the color used to fill the ZoomControl elements.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get fill() {}
set fill(value) {}
```

Property Value

String

The color used to fill the ZoomControl elements.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.innerColor Property

Gets or sets the color of plus, minus and arrow icons.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get innerColor() {}
set innerColor(value) {}
```

Property Value

String

The color of plus, minus and arrow icons.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.maxZoomFactor Property

Gets or sets the maximum zoom level allowed to set through this control.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get maxZoomFactor() {}
set maxZoomFactor(value) {}
```

Property Value

Number

The maximum zoom level, specified as percentage. The default value is 200.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.minZoomFactor Property

Gets or sets the minimum zoom level allowed to set through this control.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get minZoomFactor() {}
set minZoomFactor(value) {}
```

Property Value

Number

The minimum zoom level.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.padding Property

Gets or sets the padding of the control's contents.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get padding() {}
set padding(value) {}
```

Property Value

Number

The padding of the control's contents.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.scrollStep Property

Gets or sets the scroll offset added when users click the pan arrows.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get scrollStep() {}
set scrollStep(value) {}
```

Property Value

Number

The scroll step.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.shadowColor Property

Gets or sets the color of the control elements' shadow.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get shadowColor() {}
set shadowColor(value) {}
```

Property Value

String

The color of the control elements' shadow.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.showLabel Property

Gets or sets a value indicating whether the label that shows the current zoom level should be visible.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get showLabel() {}
set showLabel(value) {}
```

Property Value

Boolean

true if the label is visible, or **false** otherwise.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.snapToZoomStep Property

Gets or sets a value indicating whether the trackbar should snap to zoomStep values when dragged.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get snapToZoomStep() {}
set snapToZoomStep(value) {}
```

Property Value

Boolean

true if snapping is enabled, or **false** otherwise.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.target Property

Gets or sets the control modified by this ZoomControl.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get target() {}
set target(value) {}
```

Property Value

Control

The control modified by this ZoomControl.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.textColor Property

Gets or sets the color of the the label that shows the current zoom level.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

The color of the the label that shows the current zoom level.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.tickPosition Property

Gets or sets the current tick position of the trackbar.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get tickPosition() {}
set tickPosition(value) {}
```

Property Value

TickPosition

A member of the TickPosition enumeration.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.zoomFactor Property

Gets or sets the zoom factor.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get zoomFactor() {}
set zoomFactor(value) {}
```

Property Value

Number

The current zoom factor, specified as percentage.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl.zoomStep Property

Gets or sets the amount by which to change zoom level when + and - buttons are clicked.

Namespace: MindFusion.Controls
Assembly: ZoomControl.js

Syntax

JavaScript Copy Code

```
get zoomStep() {}
set zoomStep(value) {}
```

Property Value

Number

The zoom step. The default value is 10.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Controls Namespace

ZoomControl Events

MindFusion.Diagramming Namespace

Action Enumeration

Specifies the possible actions a user can carry out on diagram items.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
Action = {}
```

Members

	Member name	Description
	None	Do not perform any action.
	Create	Draw a new item.
	Modify	Modify an existing item.
	Split	Split a link's segment.

See Also

MindFusion.Diagramming Namespace

AddItemCommand Class

Implements creation of new diagram items.

Syntax

JavaScript Copy Code

```
// class
AddItemCommand.prototype = {}
```

Remarks

The control automatically creates instances of this class when items are added either interactively or from code.

Inheritance Hierarchy

MindFusion.Diagramming.Command
  MindFusion.Diagramming.AddItemCommand

See Also

AddItemCommand Members
MindFusion.Diagramming Namespace
RemoveItemCommand Class
ChangeItemCommand Class

AddItemCommand Members

Implements creation of new diagram items.

The following tables list the members exposed by the AddItemCommand type.

Public Constructors

	Name	Description
	AddItemCommand	Initializes a new instance of the **AddItemCommand** class.

Public Methods

	Name	Description
	execute	Command.execute override.
	redo	Command.redo override.
	undo	Command.undo override.

See Also

AddItemCommand Constructor

Initializes a new instance of the AddItemCommand class.

Namespace: MindFusion.Diagramming
Assembly: AddItemCommand.js

Syntax

JavaScript Copy Code

```
function AddItemCommand (diagram, item)
```

Parameters

diagram
The Diagram modified by this command.

item
A new DiagramItem that should be added to the **Diagram**.

See Also

AddItemCommand Class
MindFusion.Diagramming Namespace

AddItemCommand Methods

AddItemCommand.execute Method

Command.execute override.

Namespace: MindFusion.Diagramming
Assembly: AddItemCommand.js

Syntax

JavaScript Copy Code

```
function execute ()
```

See Also

AddItemCommand Members
AddItemCommand Class
MindFusion.Diagramming Namespace

AddItemCommand.redo Method

Command.redo override.

Namespace: MindFusion.Diagramming
Assembly: AddItemCommand.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

AddItemCommand Members
AddItemCommand Class
MindFusion.Diagramming Namespace

AddItemCommand.undo Method

Command.undo override.

Namespace: MindFusion.Diagramming
Assembly: AddItemCommand.js

Syntax

JavaScript Copy Code

```
function undo ()
```

See Also

AddItemCommand Members
AddItemCommand Class
MindFusion.Diagramming Namespace

AdjustmentHandles Enumeration

Identifies the distinct selection handles of a node.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
AdjustmentHandles = {}
```

Members

	Member name	Description
	None	Specifies that none of the selection handles can be used to modify the node.
	ResizeTopLeft	The top-left handle is enabled and can be used to resize the node.
	ResizeTopRight	The top-right handle is enabled and can be used to resize the node.
	ResizeBottomRight	The bottom-right handle is enabled and can be used to resize the node.
	ResizeBottomLeft	The bottom-left handle is enabled and can be used to resize the node.
	ResizeTopCenter	The top-center handle is enabled and can be used to resize the node vertically.
	ResizeMiddleRight	The middle-right handle is enabled and can be used to resize the node horizontally.
	ResizeBottomCenter	The bottom-center handle is enabled and can be used to resize the node vertically.
	ResizeMiddleLeft	The middle-left handle is enabled and can be used to resize the node horizontally.

	Move	The center selection handle is enabled and can be used to move the node.
	Rotate	The rotation handle is enabled and can be used to rotate the node.
	All	All selection handles are enabled and can be used to modify the node.

Remarks

A combination of **AdjustmentHandles** elements can be assigned to the enabledHandles property of nodes.

See Also

MindFusion.Diagramming Namespace
HandlesStyle Enumeration

AeroEffect Class

Represents an effect that adds semitransparency to the node and emphasizes its outline by adding a shade and an inner stroke.

Syntax

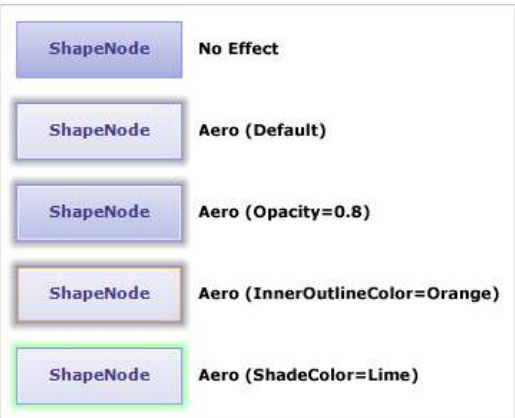
JavaScript Copy Code

```
// class
AeroEffect.prototype = {}
```

Remarks

This effect can be applied to all nodes in a Diagram by adding **AeroEffect** instance to the nodeEffects collection, or to individual nodes by adding it to their effects array.

The following image illustrates the Aero effect applied to a rectangular node as well as how the different properties affect the output:



Example

The following example demonstrates how to apply aero effect to the nodes of an existing diagram.

JavaScript Copy Code

```
var AeroEffect = MindFusion.Diagramming.AeroEffect;

var effect = new AeroEffect();
effect.opacity = 0.5;
effect.innerOutlineColor = "rgba(255,255,255,0.5)";
diagram.nodeEffects.push(effect);
```

Inheritance Hierarchy

MindFusion.Diagramming.NodeEffect
  MindFusion.Diagramming.AeroEffect

See Also

AeroEffect Members
MindFusion.Diagramming Namespace
Node Effects
GlassEffect Class

AeroEffect Members

Represents an effect that adds semitransparency to the node and emphasizes its outline by adding a shade and an inner stroke.

The following tables list the members exposed by the AeroEffect type.

Public Constructors

	Name	Description
	AeroEffect	Initializes a new instance of the AeroEffect class.

Public Properties

	Name	Description
	innerOutlineColor	Gets or sets the color of the inner outline.
	opacity	Gets or sets the opacity to apply to node's background.
	shadeColor	Gets or sets the color of the shade.

Public Methods

	Name	Description
	Clone	**ICloneable.Clone** implementation. (Inherited from NodeEffect.)

Public Events

	Name	Description
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation. (Inherited from NodeEffect.)

See Also

AeroEffect Class
MindFusion.Diagramming Namespace

AeroEffect Constructor

Initializes a new instance of the AeroEffect class.

Namespace: MindFusion.Diagramming
Assembly: NodeEffects.js

Syntax

JavaScript — Copy Code

```
function AeroEffect ()
```

See Also

AeroEffect Properties

AeroEffect.innerOutlineColor Property

Gets or sets the color of the inner outline.

Namespace: MindFusion.Diagramming
Assembly: NodeEffects.js

Syntax

JavaScript Copy Code

```
get innerOutlineColor() {}
set innerOutlineColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value. The default value is "white".

See Also

AeroEffect Members
AeroEffect Class
MindFusion.Diagramming Namespace

AeroEffect.opacity Property

Gets or sets the opacity to apply to node's background.

Namespace: MindFusion.Diagramming
Assembly: NodeEffects.js

Syntax

JavaScript Copy Code

```
get opacity() {}
set opacity(value) {}
```

Property Value

Number

A floating point value between 0 and 1.

See Also

AeroEffect Members
AeroEffect Class
MindFusion.Diagramming Namespace

AeroEffect.shadeColor Property

Gets or sets the color of the shade.

Namespace: MindFusion.Diagramming
Assembly: NodeEffects.js

Syntax

JavaScript

```
get shadeColor() {}
set shadeColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value. The default value is "black".

See Also

AeroEffect Members
AeroEffect Class
MindFusion.Diagramming Namespace

AeroEffect Events

NodeEffect.PropertyChanged Event

INotifyPropertyChanged.PropertyChanged implementation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public event PropertyChangedEventHandler PropertyChanged
```

Visual Basic

```
Public Event PropertyChanged As PropertyChangedEventHandler
```

See Also

NodeEffect Members
NodeEffect Class
MindFusion.Diagramming Namespace

Alignment Enumeration

Specifies the alignment of text relative to its layout rectangle.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript

```
// enum
Alignment = {}
```

Members

	Member name	Description
	Near	The text is drawn in the near corner of the layout rectangle.
	Center	The text is drawn in the center of the layout rectangle.

	Far	The text is drawn in the far corner of the layout rectangle.

Remarks

Members of this enumeration can be assigned to textAlignment and lineAlignment properties of diagram items.

See Also

MindFusion.Diagramming Namespace

AnchorPattern Class

The **AnchorPattern** class represents a set of anchor points, which specify the exact locations where links are allowed to connect to nodes.

Syntax

JavaScript Copy Code

```
// class
AnchorPattern.prototype = {}
```

Remarks

A pattern can be assigned to nodes by setting their anchorPattern property. The **AnchorPattern** class exposes several predefined patterns as static fields: decision2In2Out, decision1In3Out, leftInRightOut, topInBottomOut.

Inheritance Hierarchy

MindFusion.Diagramming.AnchorPattern

See Also

AnchorPattern Members
MindFusion.Diagramming Namespace
AnchorPoint Class

AnchorPattern Members

The AnchorPattern class represents a set of anchor points, which specify the exact locations where links are allowed to connect to nodes.

The following tables list the members exposed by the **AnchorPattern** type.

Public Constructors

	Name	Description
	AnchorPattern	Initializes a new instance of the **AnchorPattern** class.

Public Fields

	Name	Description
	decision1In3Out	Gets the predefined anchor pattern decision1In3Out.
	decision2In2Out	Gets the predefined anchor pattern decision2In2Out.
	leftInRightOut	Gets the predefined anchor pattern leftInRightOut.
	topInBottomOut	Gets the predefined anchor pattern topInBottomOut.

Public Properties

	Name	Description
	id	Gets the unique identifier of this AnchorPattern.
	points	Gets or sets the anchor points in this pattern.

Public Methods

	Name	Description
	fromId	Returns a reference to the anchor pattern with the specified string identifier.

See Also

AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Fields

AnchorPattern.decision1In3Out Field

Gets the predefined anchor pattern decision1In3Out.

Namespace: MindFusion.Diagramming
Assembly: AnchorPattern.js

Syntax

JavaScript Copy Code

```
var decision1In3Out
```

Remarks

Incoming links connect to the top vertex; outgoing links can start from the other vertices.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern.decision2In2Out Field

Gets the predefined anchor pattern decision2In2Out.

Namespace: MindFusion.Diagramming
Assembly: AnchorPattern.js

Syntax

JavaScript Copy Code

```
var decision2In2Out
```

Remarks

Incoming links can connect with the top and left vertices; outgoing links can start from the right and bottom vertices.

See Also

AnchorPattern Members
AnchorPattern Class

AnchorPattern.leftInRightOut Field

Gets the predefined anchor pattern leftInRightOut.

Namespace: MindFusion.Diagramming
Assembly: AnchorPattern.js

Syntax

JavaScript Copy Code

```
var leftInRightOut
```

Remarks

The middle point of a node's bounding rectangle left side is anchor for incoming links; that on the right side - for outgoing.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern.topInBottomOut Field

Gets the predefined anchor pattern topInBottomOut.

Namespace: MindFusion.Diagramming
Assembly: AnchorPattern.js

Syntax

JavaScript Copy Code

```
var topInBottomOut
```

Remarks

The middle point of a node's bounding rectangle top side is anchor for incoming links; that on the bottom - for outgoing.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Constructor

Initializes a new instance of the AnchorPattern class.

Namespace: MindFusion.Diagramming
Assembly: AnchorPattern.js

Syntax

JavaScript Copy Code

```
function AnchorPattern (points, [patternId])
```

Parameters

points
Contains the initial set of AnchorPoint instances for the new pattern.

patternId
Optional.
String. An unique identifier for this anchor pattern.

See Also

AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Methods

AnchorPattern.fromId Method

Returns a reference to the anchor pattern with the specified string identifier.

Namespace: MindFusion.Diagramming
Assembly: AnchorPattern.js

Syntax

JavaScript Copy Code

```
function fromId (id)
```

Parameters

id
A string identifying an AnchorPattern instance.

Return Value

AnchorPattern. The AnchorPattern object whose Id is equal to the id parameter.

Remarks

The id argument should match an identifier passed to the constructor of an existing AnchorPattern object.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Properties

AnchorPattern.id Property

Gets the unique identifier of this AnchorPattern.

Namespace: MindFusion.Diagramming
Assembly: AnchorPattern.js

Syntax

JavaScript Copy Code

```
get id() {}
```

Property Value

String

A string containing the AnchorPattern id.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern.points Property

Gets or sets the anchor points in this pattern.

Namespace: MindFusion.Diagramming
Assembly: AnchorPattern.js

Syntax

JavaScript Copy Code

```
get points() {}
set points(value) {}
```

Property Value

Array

An array of AnchorPoint objects.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPoint Class

The **AnchorPoint** class represents a location in a node to which links are allowed to connect.

Syntax

JavaScript Copy Code

```
// class
AnchorPoint.prototype = {}
```

Remarks

Anchor points can be assigned to nodes only as constituent elements of an AnchorPattern object.

An anchor point can accept incoming or outgoing links, or both, as specified via the allowIncoming and allowOutgoing properties. The position of the anchor point inside a node's bounding rectangle is set through the x and y properties, expressed as percentage of rectangle's width and height.

How anchor points are marked on the display can be set using markStyle and color properties.

Inheritance Hierarchy

MindFusion.Diagramming.AnchorPoint

See Also

AnchorPoint Members
MindFusion.Diagramming Namespace
AnchorPattern Class

AnchorPoint Members

The AnchorPoint class represents a location in a node to which links are allowed to connect.

The following tables list the members exposed by the **AnchorPoint** type.

Public Constructors

	Name	Description
	AnchorPoint	Initializes a new instance of the AnchorPoint class.

Public Properties

	Name	Description
	allowIncoming	Gets or sets a value indicating whether the anchor point accepts incoming connections.
	allowOutgoing	Gets or sets a value indicating whether the anchor point accepts outgoing connections.
	color	Gets or sets the color of the anchor point mark.
	column	Gets or sets the column index of a cell with which the anchor point is associated.
	markStyle	Gets or sets the appearance of the anchor point mark.
	size	Gets or sets the size of the anchor point mark.
	tag	Gets or sets custom data tag associated with this anchor point.
	toolTip	Gets or sets tooltip text that should be displayed when the mouse hovers over an anchor point mark.
	x	Gets or sets the horizontal position of the anchor point expressed as percentage of a node's width.
	y	Gets or sets the vertical position of the anchor point expressed as percentage of a node's height.

See Also

AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Constructor

Initializes a new instance of the AnchorPoint class.

Namespace: MindFusion.Diagramming
Assembly: AnchorPoint.js

Syntax

JavaScript Copy Code

```
function AnchorPoint (x, y, [allowIncoming, [allowOutgoing, [style, [color, [size]]]]])
```

Parameters

x
Number. The horizontal position of the anchor expressed as percent of the width of a node.

y
Number. The vertical position of the anchor expressed as percent of the height of a node.

allowIncoming
Optional.
Boolean. Specifies whether incoming links can connect to this anchor.

allowOutgoing
Optional.
Boolean. Specifies whether outgoing links can connect to this anchor.

style

Optional.
MarkStyle. Defines how the anchor point is marked visually.

color

Optional.
String. Defines the color with which the anchor mark is painted.

size

Optional.
Number. Defines the size of the anchor mark.

See Also

AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Properties

AnchorPoint.allowIncoming Property

Gets or sets a value indicating whether the anchor point accepts incoming connections.

Namespace: MindFusion.Diagramming
Assembly: AnchorPoint.js

Syntax

JavaScript Copy Code

```
get allowIncoming() {}
set allowIncoming(value) {}
```

Property Value

Boolean

true to accept incoming connections, or **false** otherwise.

Remarks

If no incoming anchor points are defined for a node, link end points can be placed anywhere inside that node.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace
allowOutgoing Property
validateAnchorPoint Event

AnchorPoint.allowOutgoing Property

Gets or sets a value indicating whether the anchor point accepts outgoing connections.

Namespace: MindFusion.Diagramming
Assembly: AnchorPoint.js

Syntax

JavaScript Copy Code

```
get allowOutgoing() {}
set allowOutgoing(value) {}
```

Property Value

Boolean

true to accept outgoing connections, or **false** otherwise.

Remarks

If no outgoing anchor points are defined for a node, link start points can be placed anywhere inside that node.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace
allowIncoming Property
validateAnchorPoint Event

AnchorPoint.color Property

Gets or sets the color of the anchor point mark.

Namespace: MindFusion.Diagramming
Assembly: AnchorPoint.js

Syntax

JavaScript Copy Code

```
get color() {}
set color(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

Remarks

The anchor mark shape specified via MarkStyle is painted with this color.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.column Property

Gets or sets the column index of a cell with which the anchor point is associated.

Namespace: MindFusion.Diagramming
Assembly: AnchorPoint.js

Syntax

JavaScript Copy Code

```
get column() {}
set column(value) {}
```

Property Value

Number

The column index of a cell with which the anchor point is associated.

Remarks

If **column** is set to -1, the point relative coordinates specify percentage of row's extents. Otherwise the point coordinates specify percentage of cell's extents.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.markStyle Property

Gets or sets the appearance of the anchor point mark.

Namespace: MindFusion.Diagramming
Assembly: AnchorPoint.js

Syntax

JavaScript Copy Code

```
get markStyle() {}
set markStyle(value) {}
```

Property Value

MarkStyle

A member of the MarkStyle enumeration.

Remarks

Specifies how anchor points are marked visually on the screen.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.size Property

Gets or sets the size of the anchor point mark.

Namespace: MindFusion.Diagramming
Assembly: AnchorPoint.js

Syntax

JavaScript Copy Code

```
get size() {}
set size(value) {}
```

Property Value

Number

A number specifying the size.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.tag Property

Gets or sets custom data tag associated with this anchor point.

Namespace: MindFusion.Diagramming
Assembly: AnchorPoint.js

Syntax

JavaScript Copy Code

```
get tag() {}
set tag(value) {}
```

Property Value

Object

An Object containing custom data.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.toolTip Property

Gets or sets tooltip text that should be displayed when the mouse hovers over an anchor point mark.

Namespace: MindFusion.Diagramming
Assembly: AnchorPoint.js

Syntax

JavaScript Copy Code

```
get toolTip() {}
set toolTip(value) {}
```

Property Value

String

A string containing the tooltip text.

Remarks

Tooltip text is displayed in a small popup window near the cursor when the mouse rests for a while over an anchor point.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.x Property

Gets or sets the horizontal position of the anchor point expressed as percentage of a node's width.

Namespace: MindFusion.Diagramming
Assembly: AnchorPoint.js

Syntax

JavaScript Copy Code

```
get x() {}
set x(value) {}
```

Property Value

Number

The horizontal position of the anchor point expressed as percentage of a node's width.

Remarks

Keep the property value within 0-100 range, or otherwise it will lead to unexpected behavior.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.y Property

Gets or sets the vertical position of the anchor point expressed as percentage of a node's height.

Namespace: MindFusion.Diagramming
Assembly: AnchorPoint.js

Syntax

JavaScript Copy Code

```
get y() {}
set y(value) {}
```

Property Value

Number

The vertical position of the anchor point expressed as percentage of a node's height.

Remarks

Keep the property value within 0-100 range, or otherwise it will lead to unexpected behavior.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

ArrowHeads Class

Provides acccess to predefined arrowhead shapes.

Syntax

JavaScript Copy Code

```
// class
ArrowHeads.prototype = {}
```

Remarks

This class lists the **predefined shapes** designed as arrowheads, which can be assigned to headShape or baseShape property of links.

Inheritance Hierarchy

MindFusion.Diagramming.ArrowHeads

See Also

ArrowHeads Members
MindFusion.Diagramming Namespace

ArrowHeads Members

Provides acccess to predefined arrowhead shapes.

The following tables list the members exposed by the ArrowHeads type.

Public Methods

	Name	Description
	Arrow	Gets the Arrow predefined shape.
	BackSlash	Gets the BackSlash predefined shape.
	BowArrow	Gets the BowArrow predefined shape.
	Circle	Gets the Circle predefined shape.
	DefaultFlow	Gets the DefaultFlow predefined shape.
	DoubleArrow	Gets the DoubleArrow predefined shape.
	None	Do not display an arrow shape.
	Pentagon	Gets the PentagonArrow predefined shape.
	PointerArrow	Gets the PointerArrow predefined shape.
	Quill	Gets the Quill predefined shape.
	Reversed	Gets the Reversed predefined shape.
	RevTriangle	Gets the RevTriangle predefined shape.
	RevWithCirc	Gets the RevWithCirc predefined shape.
	RevWithLine	Gets the RevWithLine predefined shape.
	Rhombus	Gets the Rhombus predefined shape.
	Slash	Gets the Slash predefined shape.
	Tetragon	Gets the Tetragon predefined shape.
	Triangle	Gets the Triangle predefined shape.

See Also

ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads Methods

ArrowHeads.Arrow Method

Gets the Arrow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript — Copy Code

```
function Arrow ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.BackSlash Method

Gets the BackSlash predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function BackSlash ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.BowArrow Method

Gets the BowArrow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function BowArrow ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Circle Method

Gets the Circle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function Circle ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.DefaultFlow Method

Gets the DefaultFlow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function DefaultFlow ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.DoubleArrow Method

Gets the DoubleArrow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function DoubleArrow ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.None Method

Do not display an arrow shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function None ()
```

Return Value

Shape. A **null** reference.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Pentagon Method

Gets the PentagonArrow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function Pentagon ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.PointerArrow Method

Gets the PointerArrow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function PointerArrow ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Quill Method

Gets the Quill predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function Quill ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Reversed Method

Gets the Reversed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function Reversed ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.RevTriangle Method

Gets the RevTriangle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function RevTriangle ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.RevWithCirc Method

Gets the RevWithCirc predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function RevWithCirc ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.RevWithLine Method

Gets the RevWithLine predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function RevWithLine ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Rhombus Method

Gets the Rhombus predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript Copy Code

```
function Rhombus ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Slash Method

Gets the Slash predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript — Copy Code

```
function Slash ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Tetragon Method

Gets the Tetragon predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript — Copy Code

```
function Tetragon ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Triangle Method

Gets the Triangle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: ArrowHeads.js

Syntax

JavaScript — Copy Code

```
function Triangle ()
```

Return Value

Shape. An instance of the Shape class representing the predefined shape.

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

AutoResize Enumeration

Defines in which directions automatic resizing can enlarge the diagram scrollable area.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
AutoResize = {}
```

Members

	Member name	Description
	None	Do not enlarge the diagram area automatically.
	RightAndDown	Enlarge the diagram area to the right and down if an item is moved outside the boundaries in these directions.
	AllDirections	Enlarge the diagram area in any direction if an item is moved outside the boundaries.

Remarks

Members of this enumeration can be assigned to the autoResize property of Diagram.

See Also

MindFusion.Diagramming Namespace

Behavior Enumeration

Defines values that specify how the component responds to actions performed by the user.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
Behavior = {}
```

Members

	Member name	Description
	Modify	Objects can be selected and modified. New objects cannot be created.
	DrawShapes	Drawing with the mouse creates ShapeNode instances.
	DrawLinks	Drawing started over a node creates a DiagramLink. Otherwise a lasso rectangle is displayed allowing the selection of items.
	LinkShapes	A mode that is suitable for creating flowcharts and process diagrams. Drawing over empty document area creates a **ShapeNode** instance. Drawing started over a node creates a **DiagramLink**.
	LinkTables	A mode that is suitable for creating entity-relationship diagrams. Drawing over empty document area creates a TableNode instance. Drawing started over a node creates a **DiagramLink**.
	DrawTables	Drawing with the mouse creates **TableNode** instances.

	DrawControls	Drawing with the mouse creates ControlNode instances.
	LinkControls	Drawing over empty document area creates a **ControlNode** instance. Drawing started over a node creates a **DiagramLink**.
	DoNothing	The control ignores users actions with the mouse, but raises the appropriate mouse events. This mode allows applications to implement their own mouse-drawing behavior, disabling the default response to users actions.
	Custom	Drawing with the mouse creates instances of the type assigned to CustomNodeType.
	DrawContainers	Drawing with the mouse creates ContainerNode instances.
	LinkContainers	Drawing over empty document area creates a **ContainerNode** instance. Drawing started over a node creates a **DiagramLink**.
	Pan	Dragging with depressed mouse button pans the view.
	DrawTreeViews	Drawing with the mouse creates TreeViewNode instances.
	LinkTreeViews	Drawing over empty document area creates a **TreeViewNode** instance. Drawing started over a node creates a **DiagramLink**.
	DrawSvgNodes	Drawing with the mouse creates SvgNode instances.
	LinkSvgNodes	Drawing over empty document area creates an **SvgNode** instance. Drawing started over a node creates a **DiagramLink**.
	SelectOnly	Allow only selection of existing items. Modifying them or drawing new ones is disabled.
	Magnify	Dragging the mouse magnifies the view. Modifying them or drawing new ones is disabled.
	DrawFreeForms	Drawing with the mouse creates FreeFormNode instances.
	LinkFreeForms	Drawing over empty document area creates a **FreeFormNode** instance. Drawing started over a node creates a **DiagramLink**.
	DrawFreeShapes	Points from user input are shown using a **FreeFormNode** instance. Once the user finishes drawing, the **FreeFormNode** is replaced by a **ShapeNode** with a matching shape from Diagram.FreeFormTargets list.
	LinkFreeShapes	Points from user input are shown using a **FreeFormNode** instance. Once the user finishes drawing, the **FreeFormNode** is replaced by a **ShapeNode** with a matching shape from Diagram.FreeFormTargets list. Drawing started over a node creates a **DiagramLink**.
	PanAndModify	Items can be selected and modified. New items cannot be created, and dragging without grabbing an adjustment handle pans the view.
	MoveNodes	Dragging with the mouse moves the node under mouse pointer.
	Zoom	Dragging with the mouse draws a lasso tool that zooms the DiagramView to selected rectangular area.
	MoveLabels	Dragging with the mouse moves an ItemLabel at mouse pointer's location.

Remarks

Members of this enumeration can be assigned to the behavior property of **DiagramView**.

See Also

MindFusion.Diagramming Namespace

BehaviorBase Class

An abstract base class for types that define how the control reacts to users' actions with a mouse or other pointing devices.

Syntax

JavaScript — Copy Code

```
// class
BehaviorBase.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.BehaviorBase
- MindFusion.Diagramming.DoNothingBehavior
- MindFusion.Diagramming.DrawLinksBehavior
- MindFusion.Diagramming.DrawNodesBehavior
- MindFusion.Diagramming.LinkNodesBehavior
- MindFusion.Diagramming.MagnifyBehavior
- MindFusion.Diagramming.ModifyBehavior
- MindFusion.Diagramming.MoveNodesBehavior
- MindFusion.Diagramming.PanBehavior
- MindFusion.Diagramming.SelectOnlyBehavior

See Also

BehaviorBase Members
MindFusion.Diagramming Namespace

BehaviorBase Members

An abstract base class for types that define how the control reacts to users' actions with a mouse or other pointing devices.

The following tables list the members exposed by the BehaviorBase type.

Public Constructors

	Name	Description
	BehaviorBase	Overloaded.

Public Properties

	Name	Description
	diagram	Gets the target diagram of this behavior object.

Public Methods

	Name	Description
	createController	Creates a controller appropriate for current context and pointer position.

See Also

BehaviorBase Class
MindFusion.Diagramming Namespace

BehaviorBase Constructor

Overload List

Name	Description
BehaviorBase (Diagram)	Initializes a new instance of the BehaviorBase class.
BehaviorBase (DiagramView)	Initializes a new instance of the BehaviorBase class.

See Also

BehaviorBase Class
MindFusion.Diagramming Namespace

BehaviorBase Constructor (Diagram)

Initializes a new instance of the BehaviorBase class.

Namespace: MindFusion.Diagramming
Assembly: BehaviorBase.js

Syntax

JavaScript Copy Code

```
function BehaviorBase (parent)
```

Parameters

parent
The Diagram this behavior will be associated with.

See Also

BehaviorBase Constructor Overload List
BehaviorBase Class
MindFusion.Diagramming Namespace

BehaviorBase Constructor (DiagramView)

Initializes a new instance of the BehaviorBase class.

Namespace: MindFusion.Diagramming
Assembly: BehaviorBase.js

Syntax

JavaScript Copy Code

```
function BehaviorBase (parent)
```

Parameters

parent
DiagramView. The DiagramView this behavior will be associated with.

See Also

BehaviorBase Constructor Overload List
BehaviorBase Class
MindFusion.Diagramming Namespace

BehaviorBase Methods

BehaviorBase.createController Method

Creates a controller appropriate for current context and pointer position.

Namespace: MindFusion.Diagramming
Assembly: BehaviorBase.js

Syntax

JavaScript Copy Code

```
function createController (state)
```

Parameters

state
Object. An object representing current context and input event.

Return Value

SinglePointerController. An instance of SinglePointerController -derived class.

See Also

BehaviorBase Members
BehaviorBase Class
MindFusion.Diagramming Namespace

BehaviorBase Properties

BehaviorBase.diagram Property

Gets the target diagram of this behavior object.

Namespace: MindFusion.Diagramming
Assembly: BehaviorBase.js

Syntax

JavaScript Copy Code

```
get diagram() {}
set diagram(value) {}
```

Property Value

Diagram

An instance of the Diagram class.

See Also

BehaviorBase Members
BehaviorBase Class
MindFusion.Diagramming Namespace

Border Class

Represents a panel that draws a border around its child components.

Syntax

JavaScript Copy Code

```
// class
Border.prototype = {}
```

Remarks

Border objects can be used as components of CompositeNode.

Example

This part of a custom **CompositeNode**'s JSON template shows Text component inside a **Border**:

JavaScript Copy Code

```
{
    component: "Border",
    gridColumn: 0,
    name: "SurnameBorder",
    pen: "yellow",
    strokeThickness: 2,
    brush: "green",
    margin: "2, 0, 0, 0",
    children:
    [
        {
            component: "Text",
            name: "Surname",
            autoProperty: true,
            text: "",
            font: "Arial"
        }
    ]
},
```

Inheritance Hierarchy

MindFusion.Diagramming.Panel
MindFusion.Diagramming.SimplePanel
MindFusion.Diagramming.Border

See Also

Border Members
MindFusion.Diagramming Namespace

Border Members

Represents a panel that draws a border around its child components.

The following tables list the members exposed by the Border type.

Public Constructors

	Name	Description
	Border	Initializes a new instance of the Border class.

Public Fields

	Name	Description
	brush	The panel background color.
	pen	The border color.
	strokeThickness	The border thickness.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)

	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

Public Methods

	Name	Description
	getComponents	Gets the child components of this panel. (Inherited from Panel.)

See Also

Border Class
MindFusion.Diagramming Namespace

Border Fields

Border.brush Field

The panel background color.

Namespace: MindFusion.Diagramming
Assembly: Border.js

Syntax

JavaScript Copy Code

```
var brush
```

See Also

Border Members
Border Class
MindFusion.Diagramming Namespace

Border.pen Field

The border color.

Namespace: MindFusion.Diagramming
Assembly: Border.js

Syntax

JavaScript Copy Code

```
var pen
```

See Also

Border Members
Border Class
MindFusion.Diagramming Namespace

Border.strokeThickness Field

The border thickness.

Namespace: MindFusion.Diagramming
Assembly: Border.js

Syntax

JavaScript — Copy Code

```
var strokeThickness
```

See Also

Border Members
Border Class
MindFusion.Diagramming Namespace

Border Constructor

Initializes a new instance of the Border class.

Namespace: MindFusion.Diagramming
Assembly: Border.js

Syntax

JavaScript — Copy Code

```
function Border ()
```

See Also

Border Class
MindFusion.Diagramming Namespace

Builder Class

Provides support for fluent API.

Syntax

JavaScript — Copy Code

```
// class
Builder.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Builder

See Also

Builder Members
MindFusion.Diagramming Namespace

Builder Members

Provides support for fluent API.

The following tables list the members exposed by the Builder type.

Public Constructors

	Name	Description
	Builder	Initializes a new instance of the Builder class.

See Also

Builder Class
MindFusion.Diagramming Namespace

Builder Constructor

Initializes a new instance of the Builder class.

Namespace: MindFusion.Diagramming
Assembly: Builders.js

Syntax

JavaScript Copy Code

```
function Builder ()
```

See Also

Builder Class
MindFusion.Diagramming Namespace

Button Class

Represents a button component for CompositeNodes.

Syntax

JavaScript Copy Code

```
// class
Button.prototype = {}
```

Remarks

Buttons respond to user clicks by calling the JavaScript function whose name is assigned to their **clickHandler** attribute. A button can show a text label assigned to the **text** attribute, or an image whose URL is assigned to **imageLocation**.

Example

The following example adds a button to CompositeNode template:

JavaScript Copy Code

```
{
    component: "Button",
    brush: "#ccc",
    text: "Delete",
    width: 30,
    cornerRadius: 3.5,
    clickHandler: "onDeleteClick"
}
```

Inheritance Hierarchy

MindFusion.Drawing.Component
  MindFusion.Diagramming.Button

See Also

Button Members
MindFusion.Diagramming Namespace

Button Members

Represents a button component for CompositeNodes.

The following tables list the members exposed by the Button type.

Public Constructors

	Name	Description
	Button	Initializes a new instance of the Button class.

Public Fields

	Name	Description
	brush	The button background color.
	font	A MindFusion.Drawing.Font instance specifying button's font.
	height	The desired height of the button.
	pen	The border color.
	strokeThickness	The border thickness.
	textColor	The text color.
	width	The desired width of the button.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	text	Gets or sets the text caption displayed in this button.
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

See Also

Button Class
MindFusion.Diagramming Namespace

Button Fields

Button.brush Field

The button background color.

Namespace: MindFusion.Diagramming
Assembly: Button.js

Syntax

JavaScript Copy Code

```
var brush
```

See Also

Button Members
Button Class
MindFusion.Diagramming Namespace

Button.font Field

A MindFusion.Drawing.Font instance specifying button's font.

Namespace: MindFusion.Diagramming
Assembly: Button.js

Syntax

JavaScript Copy Code

```
var font
```

See Also

Button Members
Button Class
MindFusion.Diagramming Namespace

Button.height Field

The desired height of the button.

Namespace: MindFusion.Diagramming
Assembly: Button.js

Syntax

JavaScript Copy Code

```
var height
```

See Also

Button Members
Button Class
MindFusion.Diagramming Namespace

Button.pen Field

The border color.

Namespace: MindFusion.Diagramming
Assembly: Button.js

Syntax

JavaScript Copy Code

```
var pen
```

See Also

Button Members
Button Class
MindFusion.Diagramming Namespace

Button.strokeThickness Field

The border thickness.

Namespace: MindFusion.Diagramming
Assembly: Button.js

Syntax

JavaScript Copy Code

```
var strokeThickness
```

See Also

Button Members
Button Class
MindFusion.Diagramming Namespace

Button.textColor Field

The text color.

Namespace: MindFusion.Diagramming
Assembly: Button.js

Syntax

JavaScript Copy Code

```
var textColor
```

See Also

Button Members
Button Class
MindFusion.Diagramming Namespace

Button.width Field

The desired width of the button.

Namespace: MindFusion.Diagramming
Assembly: Button.js

Syntax

JavaScript Copy Code

```
var width
```

See Also

Button Members
Button Class
MindFusion.Diagramming Namespace

Button Constructor

Initializes a new instance of the Button class.

Namespace: MindFusion.Diagramming
Assembly: Button.js

Syntax

JavaScript Copy Code

```
function Button ()
```

See Also

Button Class
MindFusion.Diagramming Namespace

Button Properties

Button.text Property

Gets or sets the text caption displayed in this button.

Namespace: MindFusion.Diagramming
Assembly: Button.js

Syntax

JavaScript Copy Code

```
get text() {}
set text(value) {}
```

Property Value

String

A string containing button's caption.

See Also

Button Members
Button Class
MindFusion.Diagramming Namespace

CaptionedNode Class

CaptionedNode is an abstract base class from which the TableNode, ContainerNode and TreeViewNode classes derive.

Syntax

JavaScript Copy Code

```
// class
CaptionedNode.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.CaptionedNode
      MindFusion.Diagramming.ContainerNode
      MindFusion.Diagramming.TableNode
      MindFusion.Diagramming.TreeViewNode

See Also

CaptionedNode Members
MindFusion.Diagramming Namespace

CaptionedNode Members

CaptionedNode is an abstract base class from which the TableNode, ContainerNode and TreeViewNode classes derive.

The following tables list the members exposed by the CaptionedNode type.

Public Constructors

	Name	Description
	CaptionedNode	Initializes a new instance of the CaptionedNode class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	captionBackBrush	Gets or sets an object specifying how to fill the caption bar.
	captionFont	Gets or sets the font used to render the node's caption text.
	captionHeight	Gets or sets the height of the node's caption bar.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	dividerStroke	Gets or sets the color used to stroke the divider line between node's caption bar and content area.
	dividerStrokeDashArray	Gets or sets an array of values specifying the dash pattern of the node's divider line.
	dividerStrokeDashOffset	Gets or sets the the distance between the line start and the beginning of the dash pattern.
	dividerStrokeDashStyle	Gets or sets the line dash pattern applied when stroking node's divider line.

	dividerStrokeThickness	Gets or sets the width of the divider line between node's caption bar and content area.
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	effectiveDividerStrokeThickness	Gets the effective line thickness that should be applied when stroking this node's divider line.
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)

	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	shape	Gets or sets the shape of the node's outline.
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	clone	DiagramItem.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	fromJson	DiagramItem.fromJson override.
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)

GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
loadFromXml	DiagramItem.LoadFromXml override. Loads the node's content from an XML element.
Move	Moves the node to the specified location. (Inherited from DiagramNode.)
NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)

	Resize	Resizes the node. (Inherited from DiagramNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	restoreState	DiagramItem.restoreState override.
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveState	DiagramItem.saveState override.
	saveToXml	Saves the item content into an XML element.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	toJson	DiagramItem.toJson override.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)

	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object. (Inherited from DiagramNode.)
	LoadFromXml	DiagramItem.LoadFromXml override. (Inherited from DiagramNode.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from DiagramNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)

	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode Constructor

Initializes a new instance of the CaptionedNode class.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
function CaptionedNode ([parent])
```

Parameters

parent
Optional.
Diagram. A Diagram instance whose default node attributes are copied to this node.

See Also

CaptionedNode Methods

CaptionedNode.clone Method

DiagramItem.clone override.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript — Copy Code

```
function clone ()
```

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.fromJson Method

DiagramItem.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript — Copy Code

```
function fromJson ()
```

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.loadFromXml Method

DiagramItem.LoadFromXml override. Loads the node's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript — Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context

XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.restoreState Method

DiagramItem.restoreState override.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
function restoreState ()
```

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.saveState Method

DiagramItem.saveState override.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
function saveState ()
```

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.saveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context

XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.toJson Method

DiagramItem.toJson override.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode Properties

CaptionedNode.captionBackBrush Property

Gets or sets an object specifying how to fill the caption bar.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
get captionBackBrush() {}
set captionBackBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace
dividerStroke Property

CaptionedNode.captionFont Property

Gets or sets the font used to render the node's caption text.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
get captionFont() {}
set captionFont(value) {}
```

Property Value

Font

An object specifying the font of the caption bar.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.captionHeight Property

Gets or sets the height of the node's caption bar.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
get captionHeight() {}
set captionHeight(value) {}
```

Property Value

Number

A number specifying the height of the caption bar. The default value is 6.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.dividerStroke Property

Gets or sets the color used to stroke the divider line between node's caption bar and content area.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
get dividerStroke() {}
set dividerStroke(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

Remarks

If **dividerStroke** is not set, the divider line is drawn using the color assigned to stroke property.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace
captionBackBrush Property

CaptionedNode.dividerStrokeDashArray Property

Gets or sets an array of values specifying the dash pattern of the node's divider line.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
get dividerStrokeDashArray() {}
set dividerStrokeDashArray(value) {}
```

Property Value

Array

The pattern.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.dividerStrokeDashOffset Property

Gets or sets the the distance between the line start and the beginning of the dash pattern.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
get dividerStrokeDashOffset() {}
set dividerStrokeDashOffset(value) {}
```

Property Value

Number

The offset.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.dividerStrokeDashStyle Property

Gets or sets the line dash pattern applied when stroking node's divider line.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
get dividerStrokeDashStyle() {}
set dividerStrokeDashStyle(value) {}
```

Property Value

DashStyle

A member of the DashStyle enumeration.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.dividerStrokeThickness Property

Gets or sets the width of the divider line between node's caption bar and content area.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
get dividerStrokeThickness() {}
set dividerStrokeThickness(value) {}
```

Property Value

Number

A number specifying the stroke thickness.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.effectiveDividerStrokeThickness Property

Gets the effective line thickness that should be applied when stroking this node's divider line.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
get effectiveDividerStrokeThickness() {}
set effectiveDividerStrokeThickness(value) {}
```

Property Value

Number

A number specifying the item's divider line stroke thickness.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

CaptionedNode.shape Property

Gets or sets the shape of the node's outline.

Namespace: MindFusion.Diagramming
Assembly: CaptionedNode.js

Syntax

JavaScript Copy Code

```
get shape() {}
set shape(value) {}
```

Property Value

SimpleShape

A member of the SimpleShape enumeration.

See Also

CaptionedNode Members
CaptionedNode Class
MindFusion.Diagramming Namespace

Cell Class

Represents a cell of a TableNode.

Syntax

JavaScript Copy Code

```
// class
Cell.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Cell

See Also

Cell Members
MindFusion.Diagramming Namespace

Cell Members

Represents a cell of a TableNode.

The following tables list the members exposed by the Cell type.

Public Constructors

	Name	Description
	Cell	Initializes a new instance of the **Cell** class.

Public Properties

	Name	Description
	brush	Gets or sets the brush used to paint the cell's background.
	columnSpan	Gets or sets the number of columns spanned by this cell.

	font	Gets or sets the font used to render cell's text.
	image	Gets or sets the image displayed by this cell.
	imageAlign	Gets or sets the alignment of cell's image.
	imageContent	Gets or sets a Base64-encoded image to display inside this cell.
	imageLocation	Gets or sets URL of the image to display inside this cell.
	imagePadding	Gets or sets the padding distance between cell's borders and image.
	lineAlignment	Gets or sets the vertical alignment of cell's text.
	rowSpan	Gets or sets the number of rows spanned by this cell.
	tag	Gets or sets custom data tag associated with this cell.
	text	Gets or sets the text label displayed by this cell.
	textAlignment	Gets or sets the horizontal alignment of cell's text.
	textColor	Gets or sets the cell's text color.
	textPadding	Gets or sets the padding distance between cell's borders and its text.
	tooltip	Gets or sets the tooltip text that should be displayed when the mouse hovers over this cell.

Public Methods

	Name	Description
	loadFromXml	Loads the cell's content from an XML element.
	saveToXml	Saves the cell's content into an XML element.

See Also

Cell Class
MindFusion.Diagramming Namespace

Cell Constructor

Initializes a new instance of the Cell class.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
function Cell ()
```

See Also

Cell Class
MindFusion.Diagramming Namespace

Cell Methods

Cell.loadFromXml Method

Loads the cell's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript — Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the cell's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.saveToXml Method

Saves the cell's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript — Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the cell's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell Properties

Cell.brush Property

Gets or sets the brush used to paint the cell's background.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

```
JavaScript
get brush() {}
set brush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.columnSpan Property

Gets or sets the number of columns spanned by this cell.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

```
JavaScript
get columnSpan() {}
set columnSpan(value) {}
```

Property Value

Number

An integer value specifying the number of spanned columns.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.font Property

Gets or sets the font used to render cell's text.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

```
JavaScript
get font() {}
set font(value) {}
```

Property Value

Font

An instance of the Font class.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.image Property

Gets or sets the image displayed by this cell.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get image() {}
set image(value) {}
```

Property Value

HTMLImageElement

An image object to render as cell's content.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.imageAlign Property

Gets or sets the alignment of cell's image.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get imageAlign() {}
set imageAlign(value) {}
```

Property Value

ImageAlign

A member of the ImageAlign enumeration.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.imageContent Property

Gets or sets a Base64-encoded image to display inside this cell.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get imageContent() {}
set imageContent(value) {}
```

Property Value

String

A Base64-encoded representation of an image.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.imageLocation Property

Gets or sets URL of the image to display inside this cell.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get imageLocation() {}
set imageLocation(value) {}
```

Property Value

String

A URL string that identifies the image to be loaded.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.imagePadding Property

Gets or sets the padding distance between cell's borders and image.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get imagePadding() {}
set imagePadding(value) {}
```

Property Value

Thickness

A Thickness instance specifying image padding.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.lineAlignment Property

Gets or sets the vertical alignment of cell's text.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get lineAlignment() {}
set lineAlignment(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.rowSpan Property

Gets or sets the number of rows spanned by this cell.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get rowSpan() {}
set rowSpan(value) {}
```

Property Value

Number

An integer value specifying the number of spanned rows.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.tag Property

Gets or sets custom data tag associated with this cell.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get tag() {}
set tag(value) {}
```

Property Value

Object

An Object containing custom data.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.text Property

Gets or sets the text label displayed by this cell.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get text() {}
set text(value) {}
```

Property Value

String

A string containing the cell's text. The default value is empty string "".

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.textAlignment Property

Gets or sets the horizontal alignment of cell's text.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get textAlignment() {}
set textAlignment(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.textColor Property

Gets or sets the cell's text color.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

A string value identifying the text color.

Remarks

If this property is not set, cell's text is painted using table's cellTextColor or textColor properties.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.textPadding Property

Gets or sets the padding distance between cell's borders and its text.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get textPadding() {}
set textPadding(value) {}
```

Property Value

Thickness

A Thickness instance specifying text padding.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

Cell.tooltip Property

Gets or sets the tooltip text that should be displayed when the mouse hovers over this cell.

Namespace: MindFusion.Diagramming
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get tooltip() {}
set tooltip(value) {}
```

Property Value

String

A string containing the tooltip text.

See Also

Cell Members
Cell Class
MindFusion.Diagramming Namespace

CellEventArgs Class

Contains the arguments passed to handlers of table cell -related events.

Syntax

JavaScript Copy Code

```
// class
CellEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.CancelEventArgs
  MindFusion.Diagramming.CellEventArgs

See Also

CellEventArgs Members
MindFusion.Diagramming Namespace
Events Class

CellEventArgs Members

Contains the arguments passed to handlers of table cell -related events.

The following tables list the members exposed by the CellEventArgs type.

Public Constructors

	Name	Description
	CellEventArgs	Initializes a new instance of the CellEventArgs class.

Public Properties

	Name	Description
	cell	Gets the cell related to the event.
	column	Gets the column index of the cell related to the event.
	context	Gets the canvas rendering context.
	mouseButton	Gets which mouse button has been pressed.
	newText	Gets the text that has just been entered by the user.
	node	Gets the node related to the event.
	oldText	Gets the text that had been displayed in the node before the user edited it.
	row	Gets the row index of the cell related to the event.

Public Methods

	Name	Description
	cancelDrag	Cancels the resize of the node's row or column.

See Also

CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs Constructor

Initializes a new instance of the CellEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function CellEventArgs ()
```

See Also

CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs Methods

CellEventArgs.cancelDrag Method

Cancels the resize of the node's row or column.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function cancelDrag ()
```

Remarks

Calling this method immediately stops the operation.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace
allowResizeRows Property
allowResizeColumns Property

CellEventArgs Properties

CellEventArgs.cell Property

Gets the cell related to the event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get cell() {}
```

Property Value

Cell

A Cell instance.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.column Property

Gets the column index of the cell related to the event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get column() {}
set column(value) {}
```

Property Value

Number

The column index.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.context Property

Gets the canvas rendering context.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get context() {}
set context(value) {}
```

Property Value

CanvasRenderingContext2D

A rendering context used to draw on the Canvas element.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.mouseButton Property

Gets which mouse button has been pressed.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get mouseButton() {}
set mouseButton(value) {}
```

Property Value

Number

An integer mouse button id.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.newText Property

Gets the text that has just been entered by the user.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get newText() {}
set newText(value) {}
```

Property Value

String

A string containing newly entered text.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.node Property

Gets the node related to the event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get node() {}
set node(value) {}
```

Property Value

TableNode

A TableNode instance.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.oldText Property

Gets the text that had been displayed in the node before the user edited it.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript — Copy Code

```
get oldText() {}
set oldText(value) {}
```

Property Value

String

A string containing item's original text.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.row Property

Gets the row index of the cell related to the event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript — Copy Code

```
get row() {}
set row(value) {}
```

Property Value

Number

The row index.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellFrameStyle Enumeration

Specifies the style of the cells' frame lines.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
CellFrameStyle = {}
```

Members

	Member name	Description
	None	Table cells have no border.
	Simple	The border of table cells is simple line.
	System3D	The border has 3D look.

Remarks

Members of this enumeration can be assigned to cellFrameStyle property of TableNode.

See Also

MindFusion.Diagramming Namespace

ChangeItemCommand Class

Handles undo / redo of property changes.

Syntax

JavaScript Copy Code

```
// class
ChangeItemCommand.prototype = {}
```

Remarks

The ChangeItemCommand constructor records the state of an item, and its undo and redo methods restore the item to either its old or new state.

Example

See executeCommand for an example showing how item's state is saved for later undo / redo.

Inheritance Hierarchy

MindFusion.Diagramming.Command
  MindFusion.Diagramming.ChangeItemCommand

See Also

ChangeItemCommand Members
MindFusion.Diagramming Namespace

ChangeItemCommand Members

Handles undo / redo of property changes.

The following tables list the members exposed by the ChangeItemCommand type.

Public Constructors

	Name	Description
	ChangeItemCommand	Initializes a new instance of the **ChangeItemCommand** class.

Public Methods

	Name	Description
	execute	Command.execute override.
	redo	Command.redo override.
	undo	Command.undo override.

See Also

ChangeItemCommand Class
MindFusion.Diagramming Namespace

ChangeItemCommand Constructor

Initializes a new instance of the ChangeItemCommand class.

Namespace: MindFusion.Diagramming
Assembly: ChangeItemCommand.js

Syntax

JavaScript Copy Code

```
function ChangeItemCommand (diagram, item)
```

Parameters

diagram
The Diagram modified by this command.

item
The DiagramItem that will be modified.

See Also

ChangeItemCommand Class
MindFusion.Diagramming Namespace

ChangeItemCommand Methods

ChangeItemCommand.execute Method

Command.execute override.

Namespace: MindFusion.Diagramming
Assembly: ChangeItemCommand.js

Syntax

JavaScript Copy Code

```
function execute ()
```

See Also

ChangeItemCommand Members
ChangeItemCommand Class
MindFusion.Diagramming Namespace

ChangeItemCommand.redo Method

Command.redo override.

Namespace: MindFusion.Diagramming
Assembly: ChangeItemCommand.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

ChangeItemCommand Members
ChangeItemCommand Class
MindFusion.Diagramming Namespace

ChangeItemCommand.undo Method

Command.undo override.

Namespace: MindFusion.Diagramming
Assembly: ChangeItemCommand.js

Syntax

JavaScript Copy Code

```
function undo ()
```

See Also

ChangeItemCommand Members
ChangeItemCommand Class
MindFusion.Diagramming Namespace

ColumnStyle Enumeration

Specifies how table columns' width is set.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
ColumnStyle = {}
```

Members

	Member name	Description
	FixedWidth	The columns has fixed, unchangeable width.
	AutoWidth	The columns' width changes when the table is resized.

See Also

MindFusion.Diagramming Namespace

Command Class

Represents an action that changes the diagram and whose effects can be undone.

Syntax

JavaScript — Copy Code

```
// class
Command.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Command
- MindFusion.Diagramming.AddItemCommand
- MindFusion.Diagramming.ChangeItemCommand
- MindFusion.Diagramming.CompositeCommand
- MindFusion.Diagramming.ContainerPagesCommand
- MindFusion.Diagramming.FoldContainerCommand
- MindFusion.Diagramming.PageContainerCommand
- MindFusion.Diagramming.RemoveItemCommand
- MindFusion.Diagramming.ResizeTableColumnCommand
- MindFusion.Diagramming.ResizeTableRowCommand
- MindFusion.Diagramming.ScrollTableCommand

See Also

Command Members
MindFusion.Diagramming Namespace

Command Members

Represents an action that changes the diagram and whose effects can be undone.

The following tables list the members exposed by the Command type.

Public Constructors

	Name	Description
	Command	Initializes a new instance of the **Command** class.

Public Methods

	Name	Description
	execute	Carries out an action that changes the diagram or a diagram item.
	redo	Repeats an action that has been undone.
	undo	Undoes an action, restoring the diagram to the state it was in before carrying out the action.

See Also

Command Class
MindFusion.Diagramming Namespace

Command Constructor

Initializes a new instance of the Command class.

Namespace: MindFusion.Diagramming
Assembly: Command.js

Syntax

JavaScript — Copy Code

```
function Command (diagram)
```

Parameters

diagram
The Diagram modified by this command.

See Also

Command Class
MindFusion.Diagramming Namespace

Command Methods

Command.execute Method

Carries out an action that changes the diagram or a diagram item.

Namespace: MindFusion.Diagramming
Assembly: Command.js

Syntax

JavaScript Copy Code

```
function execute ()
```

See Also

Command Members
Command Class
MindFusion.Diagramming Namespace

Command.redo Method

Repeats an action that has been undone.

Namespace: MindFusion.Diagramming
Assembly: Command.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

Command Members
Command Class
MindFusion.Diagramming Namespace
actionRedone Event

Command.undo Method

Undoes an action, restoring the diagram to the state it was in before carrying out the action.

Namespace: MindFusion.Diagramming
Assembly: Command.js

Syntax

JavaScript Copy Code

```
function undo ()
```

See Also

Command Members
Command Class
MindFusion.Diagramming Namespace
actionUndone Event

CompatConfig Class

Contains API compatibility settings.

Syntax

JavaScript Copy Code

```
// class
CompatConfig.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.CompatConfig

See Also

CompatConfig Members
MindFusion.Diagramming Namespace

CompatConfig Members

Contains API compatibility settings.

The following tables list the members exposed by the CompatConfig type.

Public Fields

	Name	Description
	gridLayout	Indicates whether to use CSS grid layout for diagram DOM.
	pixelThickness	Indicates whether stroke thickness values are specified in pixels or diagram units.

Public Properties

	Name	Description
	propFunctions	Gets or sets whether the control should generate get/set functionsfor properties to keep compatibility with older versions of the diagramming API.

See Also

CompatConfig Class
MindFusion.Diagramming Namespace

CompatConfig Fields

CompatConfig.gridLayout Field

Indicates whether to use CSS grid layout for diagram DOM.

Namespace: MindFusion.Diagramming
Assembly: CompatConfig.js

Syntax

JavaScript Copy Code

```
CompatConfig.gridLayout = true;
```

Remarks

This is a static field. Set it to **false** in order to use VirtualScroll mode in older browsers that don't support grid layout. Note that **false** value applies absolute positioning to the diagram's parent div, which could entail further layout changes in the html page.

See Also

CompatConfig Members
CompatConfig Class
MindFusion.Diagramming Namespace

CompatConfig.pixelThickness Field

Indicates whether stroke thickness values are specified in pixels or diagram units.

Namespace: MindFusion.Diagramming
Assembly: CompatConfig.js

Syntax

JavaScript Copy Code

```
CompatConfig.pixelThickness = false;
```

Remarks

This is a static field. Set it to **true** to make StrokeThickness of DiagramItem and Style objects regardes as pixels instead of MeasureUnit values.

StrokeThickness was changed from pixels to MeasureUnit in version 3.5.1 of the library.

See Also

CompatConfig Members
CompatConfig Class
MindFusion.Diagramming Namespace

CompatConfig Properties

CompatConfig.propFunctions Property

Gets or sets whether the control should generate get/set functionsfor properties to keep compatibility with older versions of the diagramming API.

Namespace: MindFusion.Diagramming
Assembly: CompatConfig.js

Syntax

JavaScript Copy Code

```
get propFunctions() {}
set propFunctions(value) {}
```

Property Value

Boolean

true to generate get/set function pairs, or false otherwise.

See Also

CompatConfig Members
CompatConfig Class
MindFusion.Diagramming Namespace

ComponentEventArgs Class

Contains the arguments passed to handlers of component-related events.

Syntax

JavaScript Copy Code

```
// class
ComponentEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.CancelEventArgs
  MindFusion.Diagramming.ItemEventArgs
    MindFusion.Diagramming.NodeEventArgs
      MindFusion.Diagramming.ComponentEventArgs

See Also

ComponentEventArgs Members
MindFusion.Diagramming Namespace
Events Class

ComponentEventArgs Members

Contains the arguments passed to handlers of component-related events.

The following tables list the members exposed by the ComponentEventArgs type.

Public Constructors

	Name	Description
	ComponentEventArgs	Initializes a new instance of the **ComponentEventArgs** class.

Public Properties

	Name	Description
	AdjustmentHandle	Gets or sets the adjustment handle being moved by the user. (Inherited from NodeEventArgs.)
	component	Gets the component for which the event was raised.
	Label	Gets or sets a NodeLabel related to this event. (Inherited from NodeEventArgs.)
	MouseButton	Gets which mouse button has been pressed. (Inherited from NodeEventArgs.)

	MousePosition	Gets or sets the current mouse pointer position. (Inherited from NodeEventArgs.)
	Node	Gets the node for which the event was raised. (Inherited from NodeEventArgs.)

See Also

ComponentEventArgs Class
MindFusion.Diagramming Namespace

ComponentEventArgs Constructor

Initializes a new instance of the ComponentEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function ComponentEventArgs (args)
```

Parameters

args

See Also

ComponentEventArgs Class
MindFusion.Diagramming Namespace

ComponentEventArgs Properties

ComponentEventArgs.component Property

Gets the component for which the event was raised.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get component() {}
```

Property Value

Component

An instance of the Component class.

See Also

ComponentEventArgs Members
ComponentEventArgs Class
MindFusion.Diagramming Namespace

CompositeCommand Class

Represents a set of Command instances as a single operation. Undoing or redoing the composite action, respectively undoes or redoes all its constituent subactions at the same time.

Syntax

JavaScript Copy Code

```
// class
CompositeCommand.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Command
  MindFusion.Diagramming.CompositeCommand

See Also

CompositeCommand Members
MindFusion.Diagramming Namespace

CompositeCommand Members

Represents a set of Command instances as a single operation. Undoing or redoing the composite action, respectively undoes or redoes all its constituent subactions at the same time.

The following tables list the members exposed by the CompositeCommand type.

Public Constructors

	Name	Description
	CompositeCommand	Initializes a new instance of the **CompositeCommand** class.

Public Methods

	Name	Description
	execute	Command.execute override.
	getCommands	Gets a list containing the child **Command** objects.
	redo	Command.redo override.
	undo	Command.undo override.

See Also

CompositeCommand Class
MindFusion.Diagramming Namespace

CompositeCommand Constructor

Initializes a new instance of the CompositeCommand class.

Namespace: MindFusion.Diagramming
Assembly: CompositeCommand.js

Syntax

JavaScript Copy Code

```
function CompositeCommand (diagram)
```

Parameters

diagram
The Diagram modified by this command.

See Also

CompositeCommand Class
MindFusion.Diagramming Namespace

CompositeCommand Methods

CompositeCommand.execute Method

Command.execute override.

Namespace: MindFusion.Diagramming
Assembly: CompositeCommand.js

Syntax

JavaScript Copy Code

```
function execute ()
```

See Also

CompositeCommand Members
CompositeCommand Class
MindFusion.Diagramming Namespace

CompositeCommand.getCommands Method

Gets a list containing the child Command objects.

Namespace: MindFusion.Diagramming
Assembly: CompositeCommand.js

Syntax

JavaScript Copy Code

```
function getCommands ()
```

Return Value

The list of child Command objects.

See Also

CompositeCommand Members
CompositeCommand Class
MindFusion.Diagramming Namespace

CompositeCommand.redo Method

Command.redo override.

Namespace: MindFusion.Diagramming
Assembly: CompositeCommand.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

CompositeCommand Members
CompositeCommand Class
MindFusion.Diagramming Namespace

CompositeCommand.undo Method

Command.undo override.

Namespace: MindFusion.Diagramming
Assembly: CompositeCommand.js

Syntax

JavaScript Copy Code

```
function undo ()
```

See Also

CompositeCommand Members
CompositeCommand Class
MindFusion.Diagramming Namespace

CompositeNode Class

CompositeNode implements nodes whose appearance can be defined via composition of components and layout containers.

Syntax

JavaScript Copy Code

```
// class
CompositeNode.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.CompositeNode

See Also

CompositeNode Members
MindFusion.Diagramming Namespace
Composite Nodes

CompositeNode Members

CompositeNode implements nodes whose appearance can be defined via composition of components and layout containers.

The following tables list the members exposed by the **CompositeNode** type.

Public Constructors

	Name	Description
	CompositeNode	Initializes a new instance of the **CompositeNode** class with default values supplied from the specified Diagram.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	clipToOutline	Gets or sets whether contents should clip to node's outline, when outline component is set.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)

	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)

	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
S	classFromTemplate	Generates a node class whose appearance and properties are defined using JSON template.
	clone	DiagramNode.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)

	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	fromJson	DiagramNode.fromJson override.
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	getComponent	Returns the component created for the template element with specified name.
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)

	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	loadTemplate	Load a JSON template defining node's appearance.
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	restoreState	DiagramNode.restoreState override.
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveState	DiagramNode.saveState override.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	toJson	DiagramNode.toJson override.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object. (Inherited from DiagramNode.)
	LoadFromXml	DiagramItem.LoadFromXml override. (Inherited from DiagramNode.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)

	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from DiagramNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)

	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode Constructor

Initializes a new instance of the CompositeNode class with default values supplied from the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: CompositeNode.js

Syntax

JavaScript Copy Code

```
function CompositeNode ([parent])
```

Parameters

parent
Optional.
The Diagram from which to obtain default values for the node properties.

See Also

CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode Methods

CompositeNode.classFromTemplate Method

Generates a node class whose appearance and properties are defined using JSON template.

Namespace: MindFusion.Diagramming
Assembly: CompositeNode.js

Syntax

JavaScript Copy Code

```
function classFromTemplate (className, jsonTemplate, [init])
```

Parameters

className
String. Fully qualified name of the class to generate.

jsonTemplate
Object. A JavaScript object literal containing template of node's component hierarchy.

init
Optional.
function. Optional.

Return Value

CompositeNode. A constructor function that creates instances of the new class.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.clone Method

DiagramNode.clone override.

Namespace: MindFusion.Diagramming
Assembly: CompositeNode.js

Syntax

JavaScript Copy Code

```
function clone ()
```

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.fromJson Method

DiagramNode.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: CompositeNode.js

Syntax

JavaScript Copy Code

```
function fromJson ()
```

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.getComponent Method

Returns the component created for the template element with specified name.

Namespace: MindFusion.Diagramming
Assembly: CompositeNode.js

Syntax

JavaScript Copy Code

```
function getComponent (name)
```

Parameters

name
String. A string specifying the value of the name attribute in node's template.

Return Value

Component. The component created for specified element in the template.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.loadTemplate Method

Load a JSON template defining node's appearance.

Namespace: MindFusion.Diagramming
Assembly: CompositeNode.js

Syntax

JavaScript Copy Code

```
function loadTemplate (jsonTemplate)
```

Parameters

jsonTemplate
Object. A JavaScript object literal containing template of node's component hierarchy.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.restoreState Method

DiagramNode.restoreState override.

Namespace: MindFusion.Diagramming
Assembly: CompositeNode.js

Syntax

JavaScript Copy Code

```
function restoreState ()
```

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.saveState Method

DiagramNode.saveState override.

Namespace: MindFusion.Diagramming
Assembly: CompositeNode.js

Syntax

JavaScript Copy Code

```
function saveState ()
```

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.toJson Method

DiagramNode.toJson override.

Namespace: MindFusion.Diagramming
Assembly: CompositeNode.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode Properties

CompositeNode.clipToOutline Property

Gets or sets whether contents should clip to node's outline, when outline component is set.

Namespace: MindFusion.Diagramming
Assembly: CompositeNode.js

Syntax

JavaScript Copy Code

```
get clipToOutline() {}
set clipToOutline(value) {}
```

Property Value

Boolean

true to clip contents, or **false** otherwise.

Remarks

To specify the clip path, mark a Shape component in the template with **isOutline** attribute.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeRouter Class

Implements a chain of link routers so that a link passes through the sequence until it gets routed successfully.

Syntax

JavaScript Copy Code

```
// class
CompositeRouter.prototype = {}
```

Remarks

By default diagram's linkRouter is set to a **CompositeRouter** instance that chains PatternRouter and Router. You can insert additional link routers into the routers list, including custom implementations of **Router** if target diagram should favor some specific link paths.

Inheritance Hierarchy

MindFusion.Diagramming.Router
  MindFusion.Diagramming.CompositeRouter

See Also

CompositeRouter Members
MindFusion.Diagramming Namespace

CompositeRouter Members

Implements a chain of link routers so that a link passes through the sequence until it gets routed successfully.

The following tables list the members exposed by the CompositeRouter type.

Public Constructors

	Name	Description
	CompositeRouter	Initializes a new instance of the **CompositeRouter** class.

Public Properties

	Name	Description
	avoidOverlaps	Gets or sets a value indicating whether to avoid overlapping links. (Inherited from Router.)
	diagram	Gets or sets the diagram associated with this router and all its sub-routers.
	routers	A list of router objects forming a chain through whichlinks pass until they are routed successfully.

Public Methods

	Name	Description
	route	Overrides Router.route.
	routeInternal	Overrides Router.routeInternal.
	routeLinks	Overrides Router.routeLinks.

See Also

CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter Constructor

Initializes a new instance of the CompositeRouter class.

Namespace: MindFusion.Diagramming
Assembly: CompositeRouter.js

Syntax

JavaScript Copy Code

```
function CompositeRouter ([diagram])
```

Parameters

diagram
Optional.
Diagram. The diagram this router is associated with.

See Also

CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter Methods

CompositeRouter.route Method

Overrides Router.route.

Namespace: MindFusion.Diagramming
Assembly: CompositeRouter.js

Syntax

JavaScript Copy Code

```
function route (link)
```

Parameters

link
DiagramLink. The link to route.

See Also

CompositeRouter Members
CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter.routeInternal Method

Overrides Router.routeInternal.

Namespace: MindFusion.Diagramming
Assembly: CompositeRouter.js

Syntax

JavaScript Copy Code

```
function routeInternal (link)
```

Parameters

link
DiagramLink. The link to route.

Return Value

Array.<Point>. An array of points representing the link's new shape, or null if routing fails.

See Also

CompositeRouter.routeLinks Method

Overrides Router.routeLinks.

Namespace: MindFusion.Diagramming
Assembly: CompositeRouter.js

Syntax

JavaScript Copy Code

```
function routeLinks (links)
```

Parameters

links
Array.<DiagramLink>. A list of links to route.

See Also

CompositeRouter Members
CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter Properties

CompositeRouter.diagram Property

Gets or sets the diagram associated with this router and all its sub-routers.

Namespace: MindFusion.Diagramming
Assembly: CompositeRouter.js

Syntax

JavaScript Copy Code

```
get diagram() {}
set diagram(value) {}
```

Property Value

Diagram

An instance of the Diagram class.

See Also

CompositeRouter Members
CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter.routers Property

A list of router objects forming a chain through whichlinks pass until they are routed successfully.

Namespace: MindFusion.Diagramming
Assembly: CompositeRouter.js

Syntax

JavaScript Copy Code

```
get routers() {}
set routers(value) {}
```

Property Value

Array

An array of Router -derived objects.

Remarks

By default the list contains PatternRouter and Router objects

See Also

CompositeRouter Members
CompositeRouter Class
MindFusion.Diagramming Namespace

ConnectionPoint Class

A class that represents a connection between an edge and a node in a graph. It decouples the DiagramLink class from the DiagramNode class, allowing edges to connect different types of nodes without knowing anything of their implementation details. Each **DiagramLink** object contains two **ConnectionPoint** instances describing the connections at both ends of the link.

Syntax

JavaScript Copy Code

```
// class
ConnectionPoint.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.ConnectionPoint

See Also

ConnectionPoint Members
MindFusion.Diagramming Namespace

ConnectionPoint Members

A class that represents a connection between an edge and a node in a graph. It decouples the DiagramLink class from the DiagramNode class, allowing edges to connect different types of nodes without knowing anything of their implementation details. Each **DiagramLink** object contains two ConnectionPoint instances describing the connections at both ends of the link.

The following tables list the members exposed by the **ConnectionPoint** type.

Public Constructors

	Name	Description
	ConnectionPoint	Initializes a new instance of the **ConnectionPoint** class between the specified node and link.

Public Properties

	Name	Description
	endPoint	Calculates the coordinates of this connection point expressed in document coordinates.

Public Methods

	Name	Description
	addLinkToNode	Adds the **DiagramLink** associated with this connection point to the associated **DiagramNode**.
	getInitialPoint	Returns a Point (usually the center of the bounding rectangle) contained within the associated node.
	getIntersection	Calculates the intersection point between the specified line segment and the contour of the node associated with this connection point.
	getNearestAnchorPoint	Returns the nearest anchor **Point** pertinent to this connection.
	getNearestBorderPoint	Returns the nearest point lying on the **DiagramNode** border.
	getNodeRect	Returns the bounding rectangle of the node associated with this connection point.
	linkChanges	Checks whether the link would change if attached to the specified node at the specified point.
	nodesIntersect	Checks whether the node associated with the current connection point and the node associated with the specified connection point intersect.
	removeLinkFromNode	Removes the link associated with this connection point from the associated node.
	saveEndRelative	Updates the coordinates of the connection point represented by this link, relative to the bounding rectangle of the associated node.

See Also

ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint Constructor

Initializes a new instance of the ConnectionPoint class between the specified node and link.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript Copy Code

```
function ConnectionPoint ([node, [link, [incoming]]])
```

Parameters

node
Optional. The node whose connection to a link is managed by this ConnectionPoint.

link
Optional. The link whose connection to a node is managed by this **ConnectionPoint**.

incoming
Optional. Specifies which end of the link is represented by this **ConnectionPoint**.

See Also

ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint Methods

ConnectionPoint.addLinkToNode Method

Adds the DiagramLink associated with this connection point to the associated DiagramNode.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript — Copy Code

```
function addLinkToNode ()
```

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.getInitialPoint Method

Returns a Point (usually the center of the bounding rectangle) contained within the associated node.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript — Copy Code

```
function getInitialPoint ()
```

Return Value

A Point contained within the node.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.getIntersection Method

Calculates the intersection point between the specified line segment and the contour of the node associated with this connection point.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript — Copy Code

```
function getIntersection (point1, point2)
```

Parameters

point1
The first Point of the segment.

point2
The second **point** of the segment.

Return Value

A **Point** object representing the intersection point.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.getNearestAnchorPoint Method

Returns the nearest anchor Point pertinent to this connection.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript Copy Code

```
function getNearestAnchorPoint (point)
```

Parameters

point
A Point instance specifying location in the diagram.

Return Value

AnchorPointDetails. An AnchorPointDetails specifying the anchor point properties.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.getNearestBorderPoint Method

Returns the nearest point lying on the DiagramNode border.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript Copy Code

```
function getNearestBorderPoint (point)
```

Parameters

point
A Point instance specifying location in the diagram.

Return Value

AnchorPointDetails. An AnchorPointDetails specifying the anchor point properties.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.getNodeRect Method

Returns the bounding rectangle of the node associated with this connection point.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript Copy Code

```
function getNodeRect ()
```

Return Value

The bounding rectangle of the node.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.linkChanges Method

Checks whether the link would change if attached to the specified node at the specified point.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript Copy Code

```
function linkChanges (item, point)
```

Parameters

item
A DiagramItem under the mouse cursor.

point
A point where the link will be connected.

Return Value

Boolean. true if the link would connect to a different entity if dropped at the specified location; otherwise, false.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.nodesIntersect Method

Checks whether the node associated with the current connection point and the node associated with the specified connection point intersect.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript Copy Code

```
function nodesIntersect (connectionPoint)
```

Parameters

connectionPoint
A ConnectionPoint instance.

Return Value

Boolean. true if the node associated with the current connection point and the node associated with the specified connection point intersect; otherwise, false.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.removeLinkFromNode Method

Removes the link associated with this connection point from the associated node.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript Copy Code

```
function removeLinkFromNode ()
```

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.saveEndRelative Method

Updates the coordinates of the connection point represented by this link, relative to the bounding rectangle of the associated node.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript Copy Code

```
function saveEndRelative ([ctrRelative])
```

Parameters

ctrRelative
Optional.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint Properties

ConnectionPoint.endPoint Property

Calculates the coordinates of this connection point expressed in document coordinates.

Namespace: MindFusion.Diagramming
Assembly: ConnectionPoint.js

Syntax

JavaScript Copy Code

```
get endPoint() {}
```

Property Value

Point

A Point that represents the coordinate.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionStyle Enumeration

Specifies whether links should connect to a node or its rows.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
ConnectionStyle = {}
```

Members

	Member name	Description
	Node	Links should connect to the table node when drawn interactively.
	Rows	Links should connect to one of the table's rows when drawn interactively.

Remarks

Members of this enumeration can be assigned to the connectionStyle property of TableNode or TreeViewNode.

See Also

MindFusion.Diagramming Namespace

ContainerNode Class

ContainerNode instances are diagram nodes that can contain other nodes.

Syntax

JavaScript Copy Code

```
// class
ContainerNode.prototype = {}
```

Remarks

Child node can be added to or removed from containers interactively when the allowAddChildren and allowRemoveChildren properties are enabled. Use the add method to add a node to container programmatically. Use the remove method to remove a node from the container.

A container node can be folded and unfolded, either interactively or programmatically, as specified through the foldable and folded properties. When a container is folded, it occupies a minimal area and its child nodes are hidden.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
MindFusion.Diagramming.DiagramNode
MindFusion.Diagramming.ContainerNode

See Also

ContainerNode Members
MindFusion.Diagramming Namespace

ContainerNode Members

ContainerNode instances are diagram nodes that can contain other nodes.

The following tables list the members exposed by the ContainerNode type.

Public Constructors

	Name	Description
	ContainerNode	Initializes a new instance of the ContainerNode class with default values supplied from the specified Diagram.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	allowAddChildren	Gets or sets whether users are allowed to add child nodes to the container.
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	allowRemoveChildren	Gets or sets whether users are allowed to remove child nodes from the container.
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	autoDeleteChildren	Gets or sets whether whether deleting this container should also remove its child items from the diagram.
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	captionBackBrush	Gets or sets an object specifying how to fill the caption bar.
	captionFont	Gets or sets the font used to render the node's caption text. (Inherited from CaptionedNode.)
	captionHeight	Gets or sets the height of the node's caption bar.
	children	Gets the child nodes of this container.
	clipChildren	Gets or sets whether child items should be clipped by container boundaries.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)

	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	contentPath	Creates a path corresponding to the content area rectangle, excluding caption bar.
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	dividerStroke	Gets or sets the color used to stroke the divider line between node's caption bar and content area. (Inherited from CaptionedNode.)
	dividerStrokeDashArray	Gets or sets an array of values specifying the dash pattern of the node's divider line. (Inherited from CaptionedNode.)
	dividerStrokeDashOffset	Gets or sets the the distance between the line start and the beginning of the dash pattern. (Inherited from CaptionedNode.)
	dividerStrokeDashStyle	Gets or sets the line dash pattern applied when stroking node's divider line. (Inherited from CaptionedNode.)
	dividerStrokeThickness	Gets or sets the width of the divider line between node's caption bar and content area. (Inherited from CaptionedNode.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	effectiveDividerStrokeThickness	Gets the effective line thickness that should be applied when stroking this node's divider line. (Inherited from CaptionedNode.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	enableStyledText	Gets or sets a value indicating whether styled caption text rendering is enabled.
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	foldable	Gets or sets a value indicating whether users are allowed to fold the container.
	folded	Gets or sets a value indicating whether the container is folded.
	foldedSize	Gets or sets the size of the container in folded state.
	foldIconSize	Gets or sets a value indicating the size of the fold/unfold icon.
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)

	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	shape	Gets or sets the shape of the node's outline.
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)

	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	unfoldedSize	Gets the size of the container in unfolded state.
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	add	Adds a node to this container.
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	arrange	Arranges the container's children with the given layout.
	arrangeAnimated	Arranges the container's children with the given layout, and animates items moving to their new positions.
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	clone	DiagramNode.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	containsPoint	Checks if the **ContainerNode** contains the specified point.
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	containsRecursively	Determines whether the specified node is contained within this container, either directly or inside child containers.
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)

	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	fromJson	DiagramNode.fromJson override.
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	getOutline	Gets an array of points that define the outline of this node shape.
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)

	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	hitTest	Returns the diagram item that contains the specified point.
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	loadFromXml	DiagramNode.LoadFromXml override. Loads the node's content from an XML element.
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	onRemove	DiagramItem.onRemove override.
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	remove	Removes a node from this container.
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	resizeToFitChildren	Resizes the container making it big enough to contain its child nodes.
	resizeToFitText	Makes the container node caption big enough to display its text without clipping.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	restoreState	DiagramNode.restoreState override.
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveState	DiagramNode.saveState override.
	saveToXml	Saves the node's content into an XML element.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)

	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	toJson	DiagramNode.toJson override.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	visitNodes	Visits the diagram nodes in reverse Z order.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)

	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object. (Inherited from DiagramNode.)
	LoadFromXml	DiagramItem.LoadFromXml override. (Inherited from DiagramNode.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from DiagramNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)

	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode Constructor

Initializes a new instance of the ContainerNode class with default values supplied from the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function ContainerNode ([parent])
```

Parameters

parent
Optional.
Diagram. The Diagram from which to obtain default values for the node properties.

See Also

ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode Methods

ContainerNode.add Method

Adds a node to this container.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript — Copy Code

```
function add (node)
```

Parameters

node
DiagramNode. The node that should be added to the container.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
remove Method
children Property

ContainerNode.arrange Method

Arranges the container's children with the given layout.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript — Copy Code

```
function arrange (layout)
```

Parameters

layout
Object. The Layout algorithm to apply.

Remarks

Available layout algorithms are SpringLayout, TreeLayout, BorderedTreeLayout, LayeredLayout, TreeMapLayout, FractalLayout, FlowchartLayout, TopologicalLayout, OrthogonalLayout, RadialTreeLayout, GridLayout.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
TreeLayout Class
LayeredLayout Class
SpringLayout Class

ContainerNode.arrangeAnimated Method

Arranges the container's children with the given layout, and animates items moving to their new positions.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function arrangeAnimated (layout, [duration, [animationType, [easingType]]])
```

Parameters

layout
Object. The Layout algorithm to apply.

duration
Optional.
Number. An integer, specifying the duration of the animation in milliseconds.

animationType
Optional.
A member of the AnimationType enumeration, specifying the animation type to use.

easingType
Optional.
A member of the EasingType enumeration, specifying the type of easing function to apply to the animation.

Remarks

Available layout algorithms are SpringLayout, TreeLayout, BorderedTreeLayout, LayeredLayout, TreeMapLayout, FractalLayout, FlowchartLayout, TopologicalLayout, OrthogonalLayout, RadialTreeLayout, GridLayout.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.clone Method

DiagramNode.clone override.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function clone ()
```

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.containsPoint Method

Checks if the ContainerNode contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function containsPoint (point)
```

Parameters

point
The Point to check.

Return Value

true if the point is within the bounds of this node; otherwise, **false**.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.containsRecursively Method

Determines whether the specified node is contained within this container, either directly or inside child containers.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function containsRecursively (node)
```

Parameters

node

Return Value

Boolean. **true** if the specified node is inside the container, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.fromJson Method

DiagramNode.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function fromJson ()
```

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.getOutline Method

Gets an array of points that define the outline of this node shape.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function getOutline ()
```

Return Value

An array containing the outline points.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.hitTest Method

Returns the diagram item that contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function hitTest (point, [threshold, onlyNodes, onlyInteractive])
```

Parameters

point
Point. A Point value specifying the logical coordinates of a diagram point.

threshold
Optional.
Number. Specifies the size of the area around the item outlines where the point is still considered inside the item.

onlyNodes
Boolean. true to exclude child links from the search.

onlyInteractive
Boolean. true to exclude locked items from the search.

Return Value

DiagramItem. The item that contains the specified point.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.loadFromXml Method

DiagramNode.LoadFromXml override. Loads the node's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.onRemove Method

DiagramItem.onRemove override.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function onRemove ()
```

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.remove Method

Removes a node from this container.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function remove (node)
```

Parameters

node
DiagramNode. The node that should be removed from the container.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
add Method
children Property

ContainerNode.resizeToFitChildren Method

Resizes the container making it big enough to contain its child nodes.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function resizeToFitChildren (allowShrink, margin)
```

Parameters

allowShrink
true to allow setting smaller size than current size of the container, or **false** otherwise.

margin
Specifies the margin space to leave between container's borders and child nodes.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
children Property

ContainerNode.resizeToFitText Method

Makes the container node caption big enough to display its text without clipping.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function resizeToFitText (fit, [fixPosition, [foldBtnWidth]])
```

Parameters

fit
FitSize. A member of the FitSize enumeration.

fixPosition
Optional.
Boolean. Fix the position of the container to the original location. The default is false.

foldBtnWidth
Optional.
Number. The width of the folder button, if it's not provided, the value from foldIconSize is used.

Return Value

Boolean. **true** if the caption is resized successfully, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.restoreState Method

DiagramNode.restoreState override.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function restoreState ()
```

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.saveState Method

DiagramNode.saveState override.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function saveState ()
```

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.saveToXml Method

Saves the node's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.toJson Method

DiagramNode.toJson override.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.visitNodes Method

Visits the diagram nodes in reverse Z order.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
function visitNodes (point, visitNode, relatedItem)
```

Parameters

point

visitNode

relatedItem

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode Properties

ContainerNode.allowAddChildren Property

Gets or sets whether users are allowed to add child nodes to the container.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get allowAddChildren() {}
set allowAddChildren(value) {}
```

Property Value

Boolean

true if child nodes can be added interactively, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.allowRemoveChildren Property

Gets or sets whether users are allowed to remove child nodes from the container.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get allowRemoveChildren() {}
set allowRemoveChildren(value) {}
```

Property Value

Boolean

true if child nodes can be removed interactively, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.autoDeleteChildren Property

Gets or sets whether whether deleting this container should also remove its child items from the diagram.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get autoDeleteChildren() {}
set autoDeleteChildren(value) {}
```

Property Value

Boolean

true to automatically delete child items, or **false** otherwise. The default value is **false**.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
children Property

ContainerNode.captionBackBrush Property

Gets or sets an object specifying how to fill the caption bar.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get captionBackBrush() {}
set captionBackBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
brush Property
captionHeight Property

ContainerNode.captionHeight Property

Gets or sets the height of the node's caption bar.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get captionHeight() {}
set captionHeight(value) {}
```

Property Value

Number

A number specifying the height of the caption bar. The default value is 6.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.children Property

Gets the child nodes of this container.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get children() {}
```

Property Value

Array

An array containing the child nodes of this container.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
add Method
remove Method
autoDeleteChildren Property
resizeToFitChildren Method

ContainerNode.clipChildren Property

Gets or sets whether child items should be clipped by container boundaries.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get clipChildren() {}
set clipChildren(value) {}
```

Property Value

Boolean

true if child items are clipped, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
shape Property

ContainerNode.contentPath Property

Creates a path corresponding to the content area rectangle, excluding caption bar.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get contentPath() {}
```

Property Value

Path

A Path instance.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.enableStyledText Property

Gets or sets a value indicating whether styled caption text rendering is enabled.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get enableStyledText() {}
set enableStyledText(value) {}
```

Property Value

Boolean

true to parse the node's caption text in order to find formatting tags, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.foldable Property

Gets or sets a value indicating whether users are allowed to fold the container.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get foldable() {}
set foldable(value) {}
```

Property Value

Boolean

true if the container can be folded, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
containerFolded Event
containerUnfolded Event
folded Property
foldIconSize Property
foldedSize Property

ContainerNode.folded Property

Gets or sets a value indicating whether the container is folded.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get folded() {}
set folded(value) {}
```

Property Value

Boolean

true if the container is folded, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
foldable Property
foldedSize Property

ContainerNode.foldedSize Property

Gets or sets the size of the container in folded state.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get foldedSize() {}
set foldedSize(value) {}
```

Property Value

Size

A Size instance specifying the folded size of a container.

Remarks

If this property is not set, the width of the folded container is set to its minimumSize.width, and the height to captionHeight.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
unfoldedSize Property

ContainerNode.foldIconSize Property

Gets or sets a value indicating the size of the fold/unfold icon.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get foldIconSize() {}
set foldIconSize(value) {}
```

Property Value

Number

A number specifying icon size.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
foldable Property

ContainerNode.shape Property

Gets or sets the shape of the node's outline.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript Copy Code

```
get shape() {}
set shape(value) {}
```

Property Value

SimpleShape

A member of the SimpleShape enumeration.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
cornerRadius Property

ContainerNode.unfoldedSize Property

Gets the size of the container in unfolded state.

Namespace: MindFusion.Diagramming
Assembly: ContainerNode.js

Syntax

JavaScript — Copy Code

```
get unfoldedSize() {}
```

Property Value

Size

The size of the container in unfolded state.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
foldedSize Property

ContainerPage Class

Represents a page in a PagedContainerNode.

Syntax

JavaScript — Copy Code

```
// class
ContainerPage.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.ContainerPage

See Also

ContainerPage Members
MindFusion.Diagramming Namespace

ContainerPage Members

Represents a page in a PagedContainerNode.

The following tables list the members exposed by the ContainerPage type.

Public Constructors

	Name	Description
	ContainerPage	Initializes a new instance of the ContainerPage class.

Public Properties

	Name	Description
	links	Gets the links on this page.
	nodes	Gets the nodes on this page.

	title	Gets or sets the title of this page.

Public Methods

	Name	Description
	fromJson	Deserializes the page from a JSON object.
	loadFromXml	Loads the page's content from an XML element.
	saveToXml	Saves the page's content into an XML element.
	toJson	Serializes the page to a JSON object.

See Also

ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage Constructor

Initializes a new instance of the ContainerPage class.

Namespace: MindFusion.Diagramming
Assembly: ContainerPage.js

Syntax

JavaScript Copy Code

```
function ContainerPage ([prototype])
```

Parameters

prototype
Optional.
ContainerPage. A prototype to copy.

See Also

ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage Methods

ContainerPage.fromJson Method

Deserializes the page from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: ContainerPage.js

Syntax

JavaScript Copy Code

```
function fromJson (json, context)
```

Parameters

json

context

See Also

ContainerPage Members
ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage.loadFromXml Method

Loads the page's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: ContainerPage.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process.

See Also

ContainerPage Members
ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage.saveToXml Method

Saves the page's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: ContainerPage.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process.

See Also

ContainerPage Members
ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage.toJson Method

Serializes the page to a JSON object.

Namespace: MindFusion.Diagramming
Assembly: ContainerPage.js

Syntax

JavaScript Copy Code

```
function toJson (context)
```

Parameters

context

Return Value

See Also

ContainerPage Members
ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage Properties

ContainerPage.links Property

Gets the links on this page.

Namespace: MindFusion.Diagramming
Assembly: ContainerPage.js

Syntax

JavaScript Copy Code

```
get links() {}
set links(value) {}
```

Property Value

Array

An array of DiagramLink objects.

Remarks

Unlike nodes, the control does not add links to this collection automatically, as ownership could be ambiguous (when considering nested containers, cross-container links, etc). This list lets you associate links with a page yourself when you want it hidden along with the page. The container automatically hides links between direct child nodes of non-current page.

See Also

ContainerPage Members
ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage.nodes Property

Gets the nodes on this page.

Namespace: MindFusion.Diagramming
Assembly: ContainerPage.js

Syntax

JavaScript Copy Code

```
get nodes() {}
set nodes(value) {}
```

Property Value

Array

An array of DiagramNode objects.

Remarks

PagedContainerNode automatically adds child nodes to this list when users drag them into currentPage.

See Also

ContainerPage Members
ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage.title Property

Gets or sets the title of this page.

Namespace: MindFusion.Diagramming
Assembly: ContainerPage.js

Syntax

JavaScript Copy Code

```
get title() {}
set title(value) {}
```

Property Value

String

A string containing title text.

Remarks

PagedContainerNode set its text to the title of currentPage.

See Also

ContainerPage Members
ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPagesCommand Class

Tracks changes of the pages collection of a PagedContainerNode.

Syntax

JavaScript Copy Code

```
// class
ContainerPagesCommand.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Command
  MindFusion.Diagramming.ContainerPagesCommand

See Also

ContainerPagesCommand Members
MindFusion.Diagramming Namespace

ContainerPagesCommand Members

Tracks changes of the pages collection of a PagedContainerNode.

The following tables list the members exposed by the ContainerPagesCommand type.

Public Constructors

	Name	Description
	ContainerPagesCommand	Initializes a new instance of the ContainerPagesCommand class.

Public Methods

	Name	Description
	execute	Command.execute override.
	redo	Command.redo override.
	undo	Command.undo override.

See Also

ContainerPagesCommand Class
MindFusion.Diagramming Namespace

ContainerPagesCommand Constructor

Initializes a new instance of the ContainerPagesCommand class.

Namespace: MindFusion.Diagramming
Assembly: ContainerPagesCommand.js

Syntax

JavaScript Copy Code

```
function ContainerPagesCommand (container, name)
```

Parameters

container
PagedContainerNode. The PagedContainerNode modified by this command.

name
string. The name of the command.

See Also

ContainerPagesCommand Class
MindFusion.Diagramming Namespace

ContainerPagesCommand Methods

ContainerPagesCommand.execute Method

Command.execute override.

Namespace: MindFusion.Diagramming
Assembly: ContainerPagesCommand.js

Syntax

JavaScript Copy Code

```
function execute ()
```

See Also

ContainerPagesCommand Members
ContainerPagesCommand Class
MindFusion.Diagramming Namespace

ContainerPagesCommand.redo Method

Command.redo override.

Namespace: MindFusion.Diagramming
Assembly: ContainerPagesCommand.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

ContainerPagesCommand Members
ContainerPagesCommand Class
MindFusion.Diagramming Namespace

ContainerPagesCommand.undo Method

Command.undo override.

Namespace: MindFusion.Diagramming
Assembly: ContainerPagesCommand.js

Syntax

JavaScript Copy Code

```
function undo ()
```

See Also

ContainerPagesCommand Members
ContainerPagesCommand Class
MindFusion.Diagramming Namespace

ControlNode Class

ControlNode instances are diagram nodes that render HTML content.

Syntax

JavaScript Copy Code

```
// class
ControlNode.prototype = {}
```

Remarks

ControlNode objects display custom HTML content, specified via their template property. A node's DOM structure is created inside the Content div element, which is then rendered on top of the diagram canvas. Note that this prevents

ControlNodes drawing in diagram' usual Z order, and they will always appear on top of other diagram items.

By default the diagram intercepts mouse events of all hosted html elements to enable moving nodes or drawing links. You can specify that interactive elements, such as buttons or text boxes, should receive input instead by setting their **data-interactive** attribute to **true**.

You can register event handlers for elements in the template by assigning function names to **data-event** attributes in the form data-event-'eventName'='handlerName'. Alternatively, you can attach event listeners using DOM API once nodeDomCreated is raised to signify that **content** is available. E.g. access child elements by calling args.getNode().getContent().querySelector(...) and attach handlers via elements' **addEventListener** method.

ControlNodes create DOM elements from their template only in the main diagram view, and render images instead when shown in Overview or NodeListView. Note that in this case image elements in the template will render only if specified in base64 format. The Utils.toDataUrl method helps you convert an image URL to base64 data URL.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.ControlNode

See Also

ControlNode Members
MindFusion.Diagramming Namespace

ControlNode Members

ControlNode instances are diagram nodes that render HTML content.

The following tables list the members exposed by the **ControlNode** type.

Public Constructors

	Name	Description
	ControlNode	Initializes a new instance of the **ControlNode** class with default values supplied from the specified Diagram.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)

	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)

	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	template	Gets or sets HTML string representing the DOM content rendered by this node.
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	visible	DiagramItem.visible override. Gets or sets whether this node is visible.
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	clone	DiagramNode.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	createImage	Creates the background svg image.
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	fromJson	DiagramNode.fromJson override.
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)

	getContent	Gets the content element of this node.
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	init	Returns a Builder object used to configure this ControlNode instance.
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	loadFromXml	DiagramNode.loadFromXml override. Loads the node's content from an XML element.
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	onUpdateBounds	DiagramNode.onUpdateBounds override.

	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	removeCanvasElements	DiagramNode.removeCanvasElements override.
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveToXml	Saves the node's content into an XML element.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	toJson	DiagramNode.toJson override.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
S	With	Returns a Builder object used to configure and create new **ControlNode** instances.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)

CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
LoadFromJson	Loads the node from a JSON object. (Inherited from DiagramNode.)
LoadFromXml	DiagramItem.LoadFromXml override. (Inherited from DiagramNode.)
LoadObsolete	For internal use. (Inherited from DiagramItem.)
OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
Repaint	Overloaded. (Inherited from DiagramItem.)
RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)

	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from DiagramNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

ControlNode Class
MindFusion.Diagramming Namespace

ControlNode Constructor

Initializes a new instance of the ControlNode class with default values supplied from the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

JavaScript Copy Code

```
function ControlNode ([diagramView])
```

Parameters

diagramView
Optional.
DiagramView. The DiagramView from which to obtain default values for the node properties.

See Also

ControlNode Class
MindFusion.Diagramming Namespace

ControlNode Methods

ControlNode.clone Method

DiagramNode.clone override.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

JavaScript Copy Code

```
function clone ()
```

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode.createImage Method

Creates the background svg image.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

JavaScript Copy Code

```
function createImage ()
```

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode.fromJson Method

DiagramNode.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

```
JavaScript                                                    Copy Code
function fromJson ()
```

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode.getContent Method

Gets the content element of this node.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

```
JavaScript                                                    Copy Code
function getContent ()
```

Return Value

The **div** element hosting HTML content generated from template.

Remarks

The diagram raises nodeDomCreated event once DOM elements have been created from template.

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode.init Method

Returns a Builder object used to configure this ControlNode instance.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

```
JavaScript                                                    Copy Code
function init ()
```

Return Value

A Builder instance.

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode.loadFromXml Method

DiagramNode.loadFromXml override. Loads the node's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode.onUpdateBounds Method

DiagramNode.onUpdateBounds override.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

JavaScript Copy Code

```
function onUpdateBounds ()
```

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode.removeCanvasElements Method

DiagramNode.removeCanvasElements override.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

JavaScript Copy Code

```
function removeCanvasElements ()
```

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode.saveToXml Method

Saves the node's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode.toJson Method

DiagramNode.toJson override.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode.With Method

Returns a Builder object used to configure and create new ControlNode instances.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

JavaScript Copy Code

```
function With (diagram)
```

Parameters

diagram

Return Value

A Builder instance.

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode Properties

ControlNode.template Property

Gets or sets HTML string representing the DOM content rendered by this node.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

JavaScript Copy Code

```
get template() {}
set template(value) {}
```

Property Value

String

A string containing HTML markup.

Remarks

The diagram raises nodeDomCreated event once DOM elements have been created from this template.

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

ControlNode.visible Property

DiagramItem.visible override. Gets or sets whether this node is visible.

Namespace: MindFusion.Diagramming
Assembly: ControlNode.js

Syntax

JavaScript Copy Code

```
get visible() {}
set visible(value) {}
```

Property Value

Boolean

true if this node should be displayed, or **false** otherwise.

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming Namespace

CreateLinkController Class

A controller used to draw new links in the diagram.

Syntax

JavaScript Copy Code

```
// class
CreateLinkController.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.SinglePointerController
  MindFusion.Diagramming.CreateLinkController

See Also

CreateLinkController Members
MindFusion.Diagramming Namespace

CreateLinkController Members

A controller used to draw new links in the diagram.

The following tables list the members exposed by the CreateLinkController type.

Public Constructors

	Name	Description
	CreateLinkController	Initializes a new instance of the CreateLinkController class.

Public Methods

	Name	Description
	cancel	Called when the user stops drawing or modifying a diagram element, rejecting the operation. (Inherited from SinglePointerController.)
	commit	SinglePointerController.commit override.
	drawInteraction	SinglePointerController.drawInteraction override.
	move	SinglePointerController.move override.
	start	SinglePointerController.start override.
	validate	SinglePointerController.validate override.

See Also

CreateLinkController Class
MindFusion.Diagramming Namespace

CreateLinkController Constructor

Initializes a new instance of the CreateLinkController class.

Namespace: MindFusion.Diagramming
Assembly: CreateLinkController.js

Syntax

JavaScript Copy Code

```
function CreateLinkController (diagram, link)
```

Parameters

diagram
Diagram. The parent diagram of the new link.

link
DiagramLink. The link that will be drawn by this controller.

See Also

CreateLinkController Class
MindFusion.Diagramming Namespace

CreateLinkController Methods

CreateLinkController.commit Method

SinglePointerController.commit override.

Namespace: MindFusion.Diagramming
Assembly: CreateLinkController.js

Syntax

JavaScript Copy Code

```
function commit (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

CreateLinkController Members
CreateLinkController Class
MindFusion.Diagramming Namespace

CreateLinkController.drawInteraction Method

SinglePointerController.drawInteraction override.

Namespace: MindFusion.Diagramming
Assembly: CreateLinkController.js

Syntax

JavaScript Copy Code

```
function drawInteraction (context)
```

Parameters

context
DrawingContext2D. The drawing context for diagram's underlying canvas.

See Also

CreateLinkController Members
CreateLinkController Class
MindFusion.Diagramming Namespace

CreateLinkController.move Method

SinglePointerController.move override.

Namespace: MindFusion.Diagramming
Assembly: CreateLinkController.js

Syntax

JavaScript Copy Code

```
function move (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

CreateLinkController Members
CreateLinkController Class
MindFusion.Diagramming Namespace

CreateLinkController.start Method

SinglePointerController.start override.

Namespace: MindFusion.Diagramming
Assembly: CreateLinkController.js

Syntax

JavaScript Copy Code

```
function start (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

CreateLinkController Members
CreateLinkController Class
MindFusion.Diagramming Namespace

CreateLinkController.validate Method

SinglePointerController.validate override.

Namespace: MindFusion.Diagramming
Assembly: CreateLinkController.js

Syntax

JavaScript Copy Code

```
function validate (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

Return Value

Boolean. true to accept the operation, or false otherwise.

See Also

CreateLinkController Members
CreateLinkController Class
MindFusion.Diagramming Namespace

CreateNodeController Class

A controller used to draw new nodes in the diagram.

Syntax

JavaScript Copy Code

```
// class
CreateNodeController.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.SinglePointerController
MindFusion.Diagramming.CreateNodeController

See Also

CreateNodeController Members
MindFusion.Diagramming Namespace

CreateNodeController Members

A controller used to draw new nodes in the diagram.

The following tables list the members exposed by the CreateNodeController type.

Public Constructors

	Name	Description
	CreateNodeController	Initializes a new instance of the CreateNodeController class.

Public Methods

	Name	Description
	cancel	Called when the user stops drawing or modifying a diagram element, rejecting the operation. (Inherited from SinglePointerController.)
	commit	SinglePointerController.commit override.
	drawInteraction	SinglePointerController.drawInteraction override.
	move	SinglePointerController.move override.
	start	SinglePointerController.start override.
	validate	SinglePointerController.validate override.

See Also

CreateNodeController Class
MindFusion.Diagramming Namespace

CreateNodeController Constructor

Initializes a new instance of the CreateNodeController class.

Namespace: MindFusion.Diagramming
Assembly: CreateNodeController.js

Syntax

JavaScript Copy Code

```
function CreateNodeController (diagram, node)
```

Parameters

diagram
Diagram. The parent diagram of the new node.

node
DiagramNode. The node that will be drawn by this controller.

See Also

CreateNodeController Class
MindFusion.Diagramming Namespace

CreateNodeController Methods

CreateNodeController.commit Method

SinglePointerController.commit override.

Namespace: MindFusion.Diagramming
Assembly: CreateNodeController.js

Syntax

JavaScript Copy Code

```
function commit (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

CreateNodeController Members
CreateNodeController Class
MindFusion.Diagramming Namespace

CreateNodeController.drawInteraction Method

SinglePointerController.drawInteraction override.

Namespace: MindFusion.Diagramming
Assembly: CreateNodeController.js

Syntax

JavaScript Copy Code

```
function drawInteraction (context)
```

Parameters

context
DrawingContext2D. The drawing context for diagram's underlying canvas.

See Also

CreateNodeController Members
CreateNodeController Class
MindFusion.Diagramming Namespace

CreateNodeController.move Method

SinglePointerController.move override.

Namespace: MindFusion.Diagramming
Assembly: CreateNodeController.js

Syntax

JavaScript Copy Code

```
function move (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

CreateNodeController Members
CreateNodeController Class
MindFusion.Diagramming Namespace

CreateNodeController.start Method

SinglePointerController.start override.

Namespace: MindFusion.Diagramming
Assembly: CreateNodeController.js

Syntax

JavaScript Copy Code

```
function start (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

CreateNodeController Members
CreateNodeController Class
MindFusion.Diagramming Namespace

CreateNodeController.validate Method

SinglePointerController.validate override.

Namespace: MindFusion.Diagramming
Assembly: CreateNodeController.js

Syntax

JavaScript — Copy Code

```
function validate (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

Return Value

Boolean. true to accept the operation, or false otherwise.

See Also

CreateNodeController Members
CreateNodeController Class
MindFusion.Diagramming Namespace

CreateSelectionController Class

A controller used to draw selection rectangle in the diagram.

Syntax

JavaScript — Copy Code

```
// class
CreateSelectionController.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.SinglePointerController
MindFusion.Diagramming.CreateSelectionController

See Also

CreateSelectionController Members
MindFusion.Diagramming Namespace

CreateSelectionController Members

A controller used to draw selection rectangle in the diagram.

The following tables list the members exposed by the CreateSelectionController type.

Public Constructors

	Name	Description
	CreateSelectionController	Initializes a new instance of the CreateSelectionController class.

Public Methods

	Name	Description
	cancel	Called when the user stops drawing or modifying a diagram element, rejecting the operation. (Inherited from SinglePointerController.)
	commit	SinglePointerController.commit override.
	drawInteraction	SinglePointerController.drawInteraction override.
	move	SinglePointerController.move override.

	start	SinglePointerController.start override.
	validate	Called to validate the operation at current pointer position. (Inherited from SinglePointerController.)

See Also

CreateSelectionController Class
MindFusion.Diagramming Namespace

CreateSelectionController Constructor

Initializes a new instance of the CreateSelectionController class.

Namespace: MindFusion.Diagramming
Assembly: CreateSelectionController.js

Syntax

JavaScript Copy Code

```
function CreateSelectionController (selection)
```

Parameters

selection
Selection. The selection that will be drawn by this controller.

See Also

CreateSelectionController Class
MindFusion.Diagramming Namespace

CreateSelectionController Methods

CreateSelectionController.commit Method

SinglePointerController.commit override.

Namespace: MindFusion.Diagramming
Assembly: CreateSelectionController.js

Syntax

JavaScript Copy Code

```
function commit (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

CreateSelectionController Members
CreateSelectionController Class
MindFusion.Diagramming Namespace

CreateSelectionController.drawInteraction Method

SinglePointerController.drawInteraction override.

Namespace: MindFusion.Diagramming
Assembly: CreateSelectionController.js

Syntax

```
JavaScript                                                    Copy Code
function drawInteraction (context)
```

Parameters

context
DrawingContext2D. The drawing context for diagram's underlying canvas.

See Also

CreateSelectionController Members
CreateSelectionController Class
MindFusion.Diagramming Namespace

CreateSelectionController.move Method

SinglePointerController.move override.

Namespace: MindFusion.Diagramming
Assembly: CreateSelectionController.js

Syntax

```
JavaScript                                                    Copy Code
function move (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

CreateSelectionController Members
CreateSelectionController Class
MindFusion.Diagramming Namespace

CreateSelectionController.start Method

SinglePointerController.start override.

Namespace: MindFusion.Diagramming
Assembly: CreateSelectionController.js

Syntax

```
JavaScript                                                    Copy Code
function start (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

CreateSelectionController Members
CreateSelectionController Class
MindFusion.Diagramming Namespace

DelKeyAction Enumeration

Specifies the type of action performed when the user presses the DEL key.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
DelKeyAction = {}
```

Members

	Member name	Description
	None	Indicates that nothing happens.
	DeleteActiveItem	Indicates that only the activeItem is deleted.
	DeleteSelectedItems	Indicates that all selected items are deleted.

Remarks

Members of this enumeration can be assigned to the delKeyAction property of DiagramView.

See Also

MindFusion.Diagramming Namespace

Diagram Class

The **Diagram** class represents a flow diagram.

Syntax

JavaScript — Copy Code

```
// class
Diagram.prototype = {}
```

Remarks

A diagram can contain connected nodes and links. Nodes are instances of DiagramNode-derived classes. Nodes are stored in the nodes collection. Links in MindFusion.Diagramming for JavaScript are instances of the DiagramLink class and are stored in the links collection.

Managing items

Items can be created by using the **new** operator and added to the diagram by means of the addItem method. Items can be deleted by calling the removeItem method. There are shortcut methods provided by the diagram's Factory object that can be used to create an item and add it to the diagram with just one method call, for example createShapeNode and createDiagramLink.

Controlling Users' Actions

Users actions can be validated by handling various events, fired while items are being created, modified or deleted. The allowInplaceEdit property controls whether users are allowed to edit text by double clicking on items.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramBase
  MindFusion.Diagramming.Diagram

See Also

Diagram Members

The Diagram class represents a flow diagram.

The following tables list the members exposed by the **Diagram** type.

Public Constructors

	Name	Description
	Diagram	Initializes a new instance of the Diagram class.

Public Properties

	Name	Description
	activeItem	Gets the active diagram item.
	activeItemHandlesStyle	Gets or sets the visual style of adjustment handles of the active item.
	adjustmentHandlesSize	Gets or sets the size of adjustment handles that appear around selected items.
	alignToGrid	Gets or sets a value indicating whether the diagram items should be aligned to a grid.
	allowLinksRepeat	Gets or sets a value indicating whether more than one links can connect the same origin and destination diagram nodes.
	allowMultipleResize	Gets or sets a value indicating whether multiple selected nodes can be resized simultaneously.
	allowSelfLoops	Gets or sets a value indicating whether users are allowed to draw reflexive links.
	allowSplitLinks	Gets or sets a value indicating whether link segments can be added and removed interactively.
	allowUnanchoredLinks	Gets or sets a value indicating whether users can link nodes that do not have designated anchor points.
	allowUnconnectedLinks	Gets or sets a value indicating whether users are allowed to draw links that are not connected to any node.
	autoArrangeAvoidContainers	Gets or sets a value indicating whether auto-arranged link labels should avoid containers.
	autoArrangeAvoidSegments	Gets or sets a value indicating whether auto-arranged link labels should avoid link segments.
	autoCloseDistance	Gets or sets the maximum distance between first and last points of a FreeFormNode for which the node's outline is closed automatically.
	autoDeleteLinks	Gets or sets whether links are deleted automatically when their origin or destination is deleted.
	autoResize	Gets or sets a value indicating whether the document area should be resized automatically so it fits the diagram contents.
	backBrush	Gets or sets an object specifying how to fill the diagram background.
	backgroundImageAlign	Gets or sets a value indicating how to align the background image.
	backgroundImageContent	Gets or sets a Base64-encoded image to display as diagram background.
	backgroundImageUrl	Gets or sets URL of the image to paint as diagram's background.

	crossingRadius	Gets or sets the radius length of decorations displayed at link intersection points.
	customLinkType	Gets or sets the type of DiagramLink objects to create when users starts drawing.
	customNodeType	Gets or sets the type of DiagramNode objects to create when users starts drawing.
	defaultShape	Gets or sets the default shape assigned to new ShapeNode objects.
	dirty	Gets or sets a value indicating whether the diagram has changed since loading it.
	disabledHandlesStyle	Gets or sets the visual style of disabled adjustment handles.
	dynamicLinks	Gets or sets initial value of the dynamic property of new links.
	enableSpatialIndex	Gets or sets whether to index item positions for faster hit-testing and viewport clipping queries.
	expandButtonAction	Gets or sets how the control responds to users clickingthe +/- button displayed by expandable nodes.
	expandOnIncoming	Gets or sets a value indicating the link direction in which tree branches are expanded.
	factory	Gets a Factory object that lets you programmatically add new items to the diagram.
	font	Gets or sets the default font of text displayed by diagram items.
	freeFormAttractDistance	Gets or sets the radius around dragged free-form adjustment handle in which other points of **FreeFormNode** are modified too.
	freeFormTargets	Gets or sets an array of Shape objects used to replace **FreeFormNode** instances with ShapeNode ones when behavior is set to LinkFreeShapes or **DrawFreeShapes**.
	gridColor	Gets or sets the color of grid points.
	gridOffsetX	Gets or sets the horizontal offset of the first point of the alignment grid.
	gridOffsetY	Gets or sets the vertical offset of the first point of the alignment grid.
	gridPointSize	Gets or sets the size of shapes used to represent grid points.
	gridSizeX	Gets or sets the horizontal distance between adjacent grid points.
	gridSizeY	Gets or sets the vertical distance between adjacent grid points.
	gridStyle	Gets or sets the visual style of the alignment grid.
	items	Returns an array of all items in this diagram.
	laneGrid	Gets the lane grid.
	linkBaseShape	Gets or sets the shape displayed at the base of new links.
	linkBaseShapeSize	Gets or sets the default size of arrowheads at base of new links.
	linkBrush	Gets or sets the default fill of link arrowheads.
	linkCrossings	Gets or sets a value indicating how to render the intersection points of links.
	linkHeadShape	Gets or sets the shape displayed as head of new links.
	linkHeadShapeSize	Gets or sets the default size of arrowheads.
	linkHitDistance	Gets or sets a value specifying how far from a link a click is still considered a hit.
	linkPen	Gets or sets the default stroke of links.

	linkRouter	Gets or sets the object used to find paths for auto-routed links when diagram nodes are added or their positions change.
	links	Returns an array of all links in this diagram.
	linkSegments	Gets or sets the default number of link segments.
	linkShape	Gets or sets the initial shape assigned to new links.
	linkTextStyle	Gets or sets the default text placement and orientation of new links.
	mergeThreshold	Gets or sets the maximum distance between adjacent control points of a link at which the respective segments can be merged.
	nodeEffects	Gets the global effects applied to nodes in this diagram.
	nodes	Returns an array of all nodes in this diagram.
	nodesExpandable	Gets or sets a value indicating whether newly created nodes are expandable.
	roundedLinks	Gets or sets a value indicating whether the joints between segments of Polyline and **Cascading** links should be rounded.
	roundedLinksRadius	Gets or sets the radius of joint arcs of rounded links' segments.
	routeLinks	Gets or sets initial value of the autoRoute property of new links.
	routeMargin	Gets or sets the minimal distance between nodes and auto-routed links.
	selectedItemHandlesStyle	Gets or sets the visual style of adjustment handles of selected items.
	selection	Gets a Selection instance that represents the selection of items in this diagram.
	shadowOffsetX	Gets or sets the horizontal offset of items' shadows.
	shadowOffsetY	Gets or sets the vertical offset of items' shadows.
	shadowsStyle	Gets or sets the style of shadows.
	shapeBrush	Gets or sets the default fill of **ShapeNode** objects.
	shapeLibraryLocation	Gets or sets the location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path.
	shapePen	Gets or sets the default stroke of **ShapeNode** objects.
	showAnchors	Gets or sets a value indicating when to display anchor point marks.
	showDisabledHandles	Gets or sets a value indicating whether adjustment handles are painted when they are disabled.
	showGrid	Gets or sets a value indicating whether the alignment grid is visible.
	showLaneGrid	Gets or sets a value indicating whether the lane grid is visible.
	style	Gets or sets the style associated with this diagram.
	tag	Gets or sets custom data associated with this diagram.
	theme	Gets or sets the theme associated with this diagram.
	undoEnabled	Gets or sets a value indicating whether saving action records for later undo is enabled.
	validityChecks	Gets or sets a value indicating whether validity checks are performed each time an item is added to the diagram.
	version	Gets a string identifying current version of the library.
	XmlOptions	An XmlOptions object providing properties for customizing the XML serialization process. (Inherited from DiagramBase.)

Public Methods

	Name	Description
	addItem	Adds an item to the **Diagram**.
	alignPointToGrid	Returns a point of the alignment grid nearest to the one passed as an argument.
	arrange	Arranges the **diagram** with the given layout.
	arrangeAnimated	Arranges the diagram using the given layout, and animates items moving to their new positions.
	arrangeLinkLabels	Rearranges link labels whose autoArrange property is enabled.
	cancelCompositeOperation	Stops recording undo/redo commands into a CompositeCommand and discards the records created so far.
	clearAll	Removes all items from the diagram.
	commitCompositeOperation	Stops recording undo/redo commands into a **CompositeCommand** and saves the composite in the history queue.
	executeCommand	Executes the specified command on this diagram.
	fromJson	Deserializes the diagram from a JSON string.
	fromXmlDocument	Loads diagram contents from specified XML Document.
	getContentBounds	Returns the smallest rectangle that bounds all diagram items.
	getItemAt	Finds and returns the topmost DiagramItem found at the specified location.
	getItemsAt	Finds and returns an array of the **diagram items** found at the specified location.
	getLinkAt	Finds and returns the topmost **DiagramLink** found at the specified location.
	getLinksAt	Finds and returns an array of the diagram **DiagramLink**s found at the specified location.
	getNearestNode	Returns the node nearest to the specified point.
	getNodeAt	Finds and returns the topmost **DiagramNode** found at the specified location.
	getNodesAt	Finds and returns an array of the **DiagramNode**s found at the specified location.
	isItemInteractive	Checks if the specified item can be modified interactively.
	isItemVisible	Checks if the specified item is currently visible.
	loadFromString	Loads diagram contents from specified XML or JSON string.
	loadFromXml	Loads the diagram from an XML file.
	raiseClicked	Raises an event which notifies that a Diagram item has been clicked.
	raiseCreated	Raises an event which notifies that a Diagram item has been created.
	raiseDeleted	Raises an event which notifies that a Diagram item has been deleted.
	raiseDeselected	Raises an event which notifies that a Diagram item has been deselected.
	raiseDoubleClicked	Raises an event which notifies that a Diagram item has been double-clicked.
	raiseModified	Raises an event which notifies that a Diagram item has been modified.

	RaiseNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	raisePasted	Raises an event which notifies that a Diagram item has been pasted from clipboard.
	raiseSelected	Raises an event which notifies that a Diagram item has been selected.
	raiseSelecting	Raises an event which notifies that a Diagram item will be selected.
	raiseSelectionChanged	Raises an event which notifies that the selection has changed.
	redo	Executes again an action saved in the command history.
S	registerClass	Registers a diagram item class.
S	registerComponent	Registers a component class.
S	registerItemClass	Registers a diagram item class for XML serialization support.
	removeItem	Removes the specified **item** from the **Diagram**.
	resizeToFitItem	Resize the document extents so that the specified item is contained within.
	resizeToFitItems	Resizes the document scrollable area so that it fits all diagram items.
	routeAllLinks	Routes all links in the diagram in such a way they do not cross nodes.
	saveToString	Encodes the diagram contents into an XML or JSON string.
	saveToXml	Saves the diagram to an XML file.
	startCompositeOperation	Starts recording undo/redo commands into a **CompositeCommand**.
	toJson	Serializes the diagram into a JSON string.
	toXmlDocument	Saves the diagram into an XML Document.
	TrackMouseHover	Specifies whether the diagram view should track mouse hover events. (Inherited from DiagramBase.)
	undo	Undoes an action saved in the command history.

Protected Methods

	Name	Description
	OnActionRecorded	Raises the ActionRecorded event. (Inherited from DiagramBase.)
	OnActionRecording	Raises the ActionRecording event. (Inherited from DiagramBase.)
	OnActionRedone	Raises the ActionRedone event. (Inherited from DiagramBase.)
	OnActionUndone	Raises the ActionUndone event. (Inherited from DiagramBase.)
	OnBeginLoad	Raises the BeginLoad event. (Inherited from DiagramBase.)
	OnBoundsChanged	Raises the BoundsChanged event. (Inherited from DiagramBase.)
	OnCellClicked	Raises the CellClicked event. (Inherited from DiagramBase.)
	OnCellDoubleClicked	Raises the CellDoubleClicked event. (Inherited from DiagramBase.)
	OnCellTextEdited	Raises the CellTextEdited event. (Inherited from DiagramBase.)
	OnCellTextEditing	Raises the CellTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnClicked	Raises the Clicked event. (Inherited from DiagramBase.)
	OnContainerChildAdded	Raises the ContainerChildAdded event. (Inherited from DiagramBase.)

	OnContainerChildRemoved	Raises the ContainerChildRemoved event. (Inherited from DiagramBase.)
	OnContainerFolded	Raises the ContainerFolded event. (Inherited from DiagramBase.)
	OnContainerPageChanged	Raises the ContainerPageChanged event. (Inherited from DiagramBase.)
	OnContainerUnfolded	Raises the ContainerUnfolded event. (Inherited from DiagramBase.)
	OnContentVisibilityChanged	Raises the ContentVisibilityChanged event. (Inherited from DiagramBase.)
	OnDefaultShapeChanged	Raises the DefaultShapeChanged event. (Inherited from DiagramBase.)
	OnDeserializeTag	Raises the DeserializeTag event. (Inherited from DiagramBase.)
	OnDestinationAnchorChanged	Raises the DestinationAnchorChanged event. (Inherited from DiagramBase.)
	OnDirtyChanged	Raises the DirtyChanged event. (Inherited from DiagramBase.)
	OnDoubleClicked	Raises the DoubleClicked event. (Inherited from DiagramBase.)
	OnDrawAdjustmentHandles	Raises the DrawAdjustmentHandles event. (Inherited from DiagramBase.)
	OnDrawAnchorPoint	Raises the DrawAnchorPoint event. (Inherited from DiagramBase.)
	OnDrawBackground	Raises the DrawBackground event. (Inherited from DiagramBase.)
	OnDrawBranchIndicator	Raises the DrawBranchIndicator event. (Inherited from DiagramBase.)
	OnDrawCell	Raises the DrawCell event. (Inherited from DiagramBase.)
	OnDrawForeground	Raises the DrawForeground event. (Inherited from DiagramBase.)
	OnDrawLink	Raises the DrawLink event. (Inherited from DiagramBase.)
	OnDrawNode	Raises the DrawNode event. (Inherited from DiagramBase.)
	OnEndLoad	Raises the EndLoad event. (Inherited from DiagramBase.)
	OnExpandButtonClicked	Raises the ExpandButtonClicked event. (Inherited from DiagramBase.)
	OnGroupDestroyed	Raises the GroupDestroyed event. (Inherited from DiagramBase.)
	OnHitTestAdjustmentHandles	Raises the HitTestAdjustmentHandles event. (Inherited from DiagramBase.)
	OnInitializeLasso	Raises the InitializeLasso event. (Inherited from DiagramBase.)
	OnInitializeLink	Raises the InitializeLink event. (Inherited from DiagramBase.)
	OnInitializeNode	Raises the InitializeNode event. (Inherited from DiagramBase.)
	OnItemAdded	Raises the ItemAdded event. (Inherited from DiagramBase.)
	OnItemRemoved	Raises the ItemRemoved event. (Inherited from DiagramBase.)
	OnJsonDeserializeTag	Raises the JsonDeserializeTag event. (Inherited from DiagramBase.)
	OnJsonSerializeTag	Raises the JsonSerializeTag event. (Inherited from DiagramBase.)
	OnLayerVisibilityChanged	Raises the LayerVisibilityChanged event. (Inherited from DiagramBase.)
	OnLinkActivated	Raises the LinkActivated event. (Inherited from DiagramBase.)
	OnLinkClicked	Raises the LinkClicked event. (Inherited from DiagramBase.)
	OnLinkCreateCancelled	Raises the LinkCreateCancelled event. (Inherited from DiagramBase.)
	OnLinkCreated	Raises the LinkCreated event. (Inherited from DiagramBase.)
	OnLinkCreating	Raises the LinkCreating event. (Inherited from DiagramBase.)
	OnLinkDeactivated	Raises the LinkDeactivated event. (Inherited from DiagramBase.)

	OnLinkDeleted	Raises the LinkDeleted event. (Inherited from DiagramBase.)
	OnLinkDeleting	Raises the LinkDeleting event. (Inherited from DiagramBase.)
	OnLinkDeselected	Raises the LinkDeselected event. (Inherited from DiagramBase.)
	OnLinkDoubleClicked	Raises the LinkDoubleClicked event. (Inherited from DiagramBase.)
	OnLinkHovered	Raises the LinkHovered event. (Inherited from DiagramBase.)
	OnLinkModified	Raises the LinkModified event. (Inherited from DiagramBase.)
	OnLinkModifyCancelled	Raises the LinkModifyCancelled event. (Inherited from DiagramBase.)
	OnLinkModifying	Raises the LinkModifying event. (Inherited from DiagramBase.)
	OnLinkPasted	Raises the LinkPasted event. (Inherited from DiagramBase.)
	OnLinkRouted	Raises the LinkRouted event. (Inherited from DiagramBase.)
	OnLinkSelected	Raises the LinkSelected event. (Inherited from DiagramBase.)
	OnLinkSelecting	Raises the LinkSelecting event. (Inherited from DiagramBase.)
	OnLinkSplit	Raises the LinkSplit event. (Inherited from DiagramBase.)
	OnLinkStartModifying	Raises the LinkStartModifying event. (Inherited from DiagramBase.)
	OnLinkTextEdited	Raises the LinkTextEdited event. (Inherited from DiagramBase.)
	OnLinkTextEditing	Raises the LinkTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnMeasureUnitChanged	Raises the MeasureUnitChanged event. (Inherited from DiagramBase.)
	OnNodeActivated	Raises the NodeActivated event. (Inherited from DiagramBase.)
	OnNodeClicked	Raises the NodeClicked event. (Inherited from DiagramBase.)
	OnNodeCreateCancelled	Raises the NodeCreateCancelled event. (Inherited from DiagramBase.)
	OnNodeCreated	Raises the NodeCreated event. (Inherited from DiagramBase.)
	OnNodeCreating	Raises the NodeCreating event. (Inherited from DiagramBase.)
	OnNodeDeactivated	Raises the NodeDeactivated event. (Inherited from DiagramBase.)
	OnNodeDeleted	Raises the NodeDeleted event. (Inherited from DiagramBase.)
	OnNodeDeleting	Raises the NodeDeleting event. (Inherited from DiagramBase.)
	OnNodeDeselected	Raises the NodeDeselected event. (Inherited from DiagramBase.)
	OnNodeDoubleClicked	Raises the NodeDoubleClicked event. (Inherited from DiagramBase.)
	OnNodeHovered	Raises the NodeHovered event. (Inherited from DiagramBase.)
	OnNodeModified	Raises the NodeModified event. (Inherited from DiagramBase.)
	OnNodeModifyCancelled	Raises the NodeModifyCancelled event. (Inherited from DiagramBase.)
	OnNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	OnNodePasted	Raises the NodePasted event. (Inherited from DiagramBase.)
	OnNodeSelected	Raises the NodeSelected event. (Inherited from DiagramBase.)
	OnNodeSelecting	Raises the NodeSelecting event. (Inherited from DiagramBase.)
	OnNodeStartModifying	Raises the NodeStartModifying event. (Inherited from DiagramBase.)
	OnNodeTextEdited	Raises the NodeTextEdited event. (Inherited from DiagramBase.)

	OnNodeTextEditing	Raises the NodeTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnOriginAnchorChanged	Raises the OriginAnchorChanged event. (Inherited from DiagramBase.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from DiagramBase.)
	OnQueryAlignTarget	Raises the QueryAlignTarget event. (Inherited from DiagramBase.)
	OnRepaint	Raises the Repaint event. (Inherited from DiagramBase.)
	OnRequestClipboard	Raises the RequestClipboard event. (Inherited from DiagramBase.)
	OnRequestDrawInteractions	Raises the RequestDrawInteractions event. (Inherited from DiagramBase.)
	OnRequestTimer	Raises the RequestTimer event. (Inherited from DiagramBase.)
	OnSelectionChanged	Raises the SelectionChanged event. (Inherited from DiagramBase.)
	OnSelectionModified	Raises the SelectionModified event. (Inherited from DiagramBase.)
	OnSelectionModifying	Raises the SelectionModifying event. (Inherited from DiagramBase.)
	OnSerializeTag	Raises the SerializeTag event. (Inherited from DiagramBase.)
	OnSetSelfLoopShape	Raises the SetSelfLoopShape event. (Inherited from DiagramBase.)
	OnShapeModified	Raises the ShapeModified event. (Inherited from DiagramBase.)
	OnShapeNotDefined	Raises the ShapeNotDefined event. (Inherited from DiagramBase.)
	OnTableColumnResized	Raises the TableColumnResized event. (Inherited from DiagramBase.)
	OnTableColumnResizing	Raises the TableColumnResizing event. (Inherited from DiagramBase.)
	OnTableRowResized	Raises the TableRowResized event. (Inherited from DiagramBase.)
	OnTableRowResizing	Raises the TableRowResizing event. (Inherited from DiagramBase.)
	OnTableSectionCollapsed	Raises the TableSectionCollapsed event. (Inherited from DiagramBase.)
	OnTableSectionExpanded	Raises the TableSectionExpanded event. (Inherited from DiagramBase.)
	OnTreeCollapsed	Raises the TreeCollapsed event. (Inherited from DiagramBase.)
	OnTreeExpanded	Raises the TreeExpanded event. (Inherited from DiagramBase.)
	OnTreeItemTextEditing	Raises the TreeItemTextEditing event. (Inherited from DiagramBase.)
	OnValidateAnchorPoint	Raises the ValidateAnchorPoint event. (Inherited from DiagramBase.)
	OnValidateLinkCrossing	Raises the ValidateLinkCrossing event. (Inherited from DiagramBase.)
	OnViewRemoved	Raises the ViewRemoved event. (Inherited from DiagramBase.)

Public Events

	Name	Description
	ActionRecorded	Raised when an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRecording	A validation event raised before an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRedone	Raised when an action is redone. (Inherited from DiagramBase.)
	ActionUndone	Raised when an action is undone. (Inherited from DiagramBase.)
	ActiveLayerChanged	Raised when the diagram's active layer has changed. (Inherited from DiagramBase.)

	BeginLoad	Raised when the component starts loading diagram data from a file or memory stream. (Inherited from DiagramBase.)
	BoundsChanged	Raised when the Bounds property value has changed. (Inherited from DiagramBase.)
	CellClicked	Raised when the user clicks a table's cell. (Inherited from DiagramBase.)
	CellDoubleClicked	Raised when the user double-clicks a table's cell. (Inherited from DiagramBase.)
	CellTextEdited	Raised when the user edits the text of a table's cell. (Inherited from DiagramBase.)
	CellTextEditing	A validation event that lets you approve or reject changes made to the text of a table's cell. (Inherited from DiagramBase.)
	Clicked	Raised when the user clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	ContainerChildAdded	Raised when a node is added to a ContainerNode, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerChildRemoved	Raised when a node is removed from a **ContainerNode**, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerFolded	Raised when the user folds a container by clicking the arrow-up button in the container's caption bar. (Inherited from DiagramBase.)
	ContainerPageChanged	Raised when the user changes current page of a PagedContainerNode. (Inherited from DiagramBase.)
	ContainerUnfolded	Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar. (Inherited from DiagramBase.)
	ContentVisibilityChanged	Raised when the visibility of container's content changes. (Inherited from DiagramBase.)
	DefaultShapeChanged	Raised when the DefaultShape property value has changed. (Inherited from DiagramBase.)
	DeserializeTag	Raised when the Tag of a diagram item must be deserialized from XML format. (Inherited from DiagramBase.)
	DestinationAnchorChanged	Raised when the DestinationAnchor property value of a link has changed. (Inherited from DiagramBase.)
	DirtyChanged	Raised when the **Dirty** property value has changed. (Inherited from DiagramBase.)
	DoubleClicked	Raised when the user double-clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	DrawAdjustmentHandles	Raised when item's adjustment handles must be custom-drawn. (Inherited from DiagramBase.)
	DrawAnchorPoint	Raised when an anchor point must be custom-drawn. (Inherited from DiagramBase.)
	DrawBackground	Raised to let you custom draw the diagram background. (Inherited from DiagramBase.)
	DrawBranchIndicator	Raised when a branch indicator must be custom-drawn. (Inherited from DiagramBase.)
	DrawCell	Raised when a table's cell must be custom drawn. (Inherited from DiagramBase.)
	DrawForeground	Raised to let you custom draw the diagram foreground. (Inherited from DiagramBase.)
	DrawLink	Raised when a diagram link must be custom drawn. (Inherited from DiagramBase.)

	DrawNode	Raised when a diagram node must be custom drawn. (Inherited from DiagramBase.)
	EndLoad	Raised when the component ends loading diagram data from a file or memory stream. (Inherited from DiagramBase.)
	ExpandButtonClicked	Raised when the user clicks the [±] button of an Expandable node. (Inherited from DiagramBase.)
	GroupDestroyed	Raised when a Group object is destroyed. (Inherited from DiagramBase.)
	HitTestAdjustmentHandles	Raised when custom hit testing of item's adjustment handles must be performed. (Inherited from DiagramBase.)
	InitializeLasso	Raised when the user starts drawing a selection lasso, before it is rendered on screen. (Inherited from DiagramBase.)
	InitializeLink	Raised when the user starts drawing a new link, just after the DiagramLink instance is created. (Inherited from DiagramBase.)
	InitializeNode	Raised when the user starts drawing a new node, just after the DiagramNode instance is created. (Inherited from DiagramBase.)
	ItemAdded	Raised when an item is added to the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	ItemRemoved	Raised when an item is removed from the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	JsonDeserializeTag	Raised when the **Tag** of a diagram item must be deserialized from JSON. (Inherited from DiagramBase.)
	JsonSerializeTag	Raised when the **Tag** of a diagram item must be serialized to JSON. (Inherited from DiagramBase.)
	LayerVisibilityChanged	Raised when the visibility of a Layer changes. (Inherited from DiagramBase.)
	LinkActivated	Raised when a link becomes the ActiveItem. (Inherited from DiagramBase.)
	LinkClicked	Raised when the user clicks a link. (Inherited from DiagramBase.)
	LinkCreateCancelled	Raised when the user cancels link creation. (Inherited from DiagramBase.)
	LinkCreated	Raised when the user draws a new link. (Inherited from DiagramBase.)
	LinkCreating	A validation event raised while the user is drawing a new link. (Inherited from DiagramBase.)
	LinkDeactivated	Raised when a link is no longer the **ActiveItem**. (Inherited from DiagramBase.)
	LinkDeleted	Raised when a link is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDeleting	Raised when the user tries to delete a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkDeselected	Raised when a link is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDoubleClicked	Raised when the user double-clicks a link. (Inherited from DiagramBase.)
	LinkHovered	Raised when the mouse pointer hovers over a link. (Inherited from DiagramBase.)
	LinkModified	Raised when the user moves any control point of a link. (Inherited from DiagramBase.)
	LinkModifyCancelled	Raised when the user cancels move or resize operation being done on a link. (Inherited from DiagramBase.)
	LinkModifying	A validation event raised while the user is modifying a link. (Inherited from DiagramBase.)

	LinkPasted	Raised when a link has been pasted from the clipboard. (Inherited from DiagramBase.)
	LinkRouted	Raised when a link is routed by the auto-routing algorithm. (Inherited from DiagramBase.)
	LinkSelected	Raised when a link is selected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkSelecting	Raised when the user tries to select a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkSplit	Raised when the user splits a link interactively. (Inherited from DiagramBase.)
	LinkStartModifying	A validation event raised when the user starts modifying a link. (Inherited from DiagramBase.)
	LinkTextEdited	Raised when the user edits the text of a link. (Inherited from DiagramBase.)
	LinkTextEditing	A validation event that lets you approve or reject changes made to the text of a link. (Inherited from DiagramBase.)
	MeasureUnitChanged	Raised when the MeasureUnit property value has changed. (Inherited from DiagramBase.)
	NodeActivated	Raised when a node becomes the **ActiveItem**. (Inherited from DiagramBase.)
	NodeClicked	Raised when the user clicks a node. (Inherited from DiagramBase.)
	NodeCreateCancelled	Raised when the user cancels node creation. (Inherited from DiagramBase.)
	NodeCreated	Raised when the user draws a new node. (Inherited from DiagramBase.)
	NodeCreating	A validation event raised while the user is drawing a new node. (Inherited from DiagramBase.)
	NodeDeactivated	Raised when a node is no longer the **ActiveItem**. (Inherited from DiagramBase.)
	NodeDeleted	Raised when a node is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDeleting	Raised when the user tries to delete a node, this event lets you cancel the operation. (Inherited from DiagramBase.)
	NodeDeselected	Raised when a node is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDoubleClicked	Raised when the user double-clicks a node. (Inherited from DiagramBase.)
	NodeHovered	Raised when the mouse pointer hovers over a node. (Inherited from DiagramBase.)
	NodeModified	Raised when the user moves or resizes a node. (Inherited from DiagramBase.)
	NodeModifyCancelled	Raised when the user cancels move or resize operation being done on a node. (Inherited from DiagramBase.)
	NodeModifying	A validation event raised while the user is moving or resizing a node. (Inherited from DiagramBase.)
	NodePasted	Raised when a node has been pasted from the clipboard. (Inherited from DiagramBase.)
	NodeSelected	Raised when a node is selected, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeSelecting	Raised when the user tries to select a node, this event lets you cancel the operation. (Inherited from DiagramBase.)

	NodeStartModifying	A validation event raised when the user starts modifying a node. (Inherited from DiagramBase.)
	NodeTextEdited	Raised when the user edits the text of a node. (Inherited from DiagramBase.)
	NodeTextEditing	A validation event that lets you approve or reject changes made to the text of a node. (Inherited from DiagramBase.)
	OriginAnchorChanged	Raised when the OriginAnchor property value of a link has changed. (Inherited from DiagramBase.)
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation. (Inherited from DiagramBase.)
	QueryAlignTarget	Raised to validate a node as target for alignment guides when AutoAlignNodes is enabled. (Inherited from DiagramBase.)
	Repaint	Raised when the diagram views should be repainted. (Inherited from DiagramBase.)
	RequestCacheReset	Raised when an object within the diagram requests a reset of the cached bitmap. (Inherited from DiagramBase.)
	RequestClipboard	Raised when an object within the diagram requests an operation with the clipboard. (Inherited from DiagramBase.)
	RequestDrawInteractions	For internal use. (Inherited from DiagramBase.)
	RequestTimer	Raised when an object within the diagram needs to start or stop a timer. (Inherited from DiagramBase.)
	SelectionChanged	Raised when an item is added to or removed from the selection (Inherited from DiagramBase.)
	SelectionModified	Raised when the user moves multiple selected items. (Inherited from DiagramBase.)
	SelectionModifying	Raised while the user is moving multiple selected items, this event lets you cancel the modification. (Inherited from DiagramBase.)
	SerializeTag	Raised when the **Tag** of a diagram item must be serialized into an XML document. (Inherited from DiagramBase.)
	SetSelfLoopShape	Raised when a link becomes a self-loop, giving you a chance to override the default look shape. (Inherited from DiagramBase.)
	ShapeModified	Raised when the user modifies the shape of a ShapeNode by moving one of the control points corresponding to shape formula parameters. (Inherited from DiagramBase.)
	ShapeNotDefined	Raised when the diagram being loaded contains nodes with an undefined shape. (Inherited from DiagramBase.)
	TableColumnResized	Raised when the user finishes resizing a column in a table. (Inherited from DiagramBase.)
	TableColumnResizing	Raised when the user resizes a column in a table interactively. (Inherited from DiagramBase.)
	TableRowResized	Raised when the user finishes resizing a row in a table. (Inherited from DiagramBase.)
	TableRowResizing	Raised when the user resizes a row in a table interactively. (Inherited from DiagramBase.)
	TableSectionCollapsed	Raised when the user collapses the rows of a table's section by clicking the [-] button in the section's header row. (Inherited from DiagramBase.)
	TableSectionExpanded	Raised when the user expands the rows of a table's section by clicking the [+] button in the section's header row. (Inherited from DiagramBase.)
	TreeCollapsed	Raised when a user collapses a tree branch by clicking the [-] button of an **Expandable** node. (Inherited from DiagramBase.)

	TreeExpanded	Raised when a user expands a tree branch by clicking the [+] button of an **Expandable** node. (Inherited from DiagramBase.)
	TreeItemTextEditing	A validation event that lets you approve or reject changes made to the text of a TreeViewNode's item. (Inherited from DiagramBase.)
	ValidateAnchorPoint	A validation event that lets you approve or reject connecting a link to an anchor point. (Inherited from DiagramBase.)
	ValidateLinkCrossing	A validation event that lets you approve or reject displaying a decoration at the intersection point of links. (Inherited from DiagramBase.)
	ViewRemoved	Raised when a DiagramView is detached from this diagram. (Inherited from DiagramBase.)

See Also

Diagram Class
MindFusion.Diagramming Namespace

Diagram Constructor

Initializes a new instance of the Diagram class.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function Diagram ()
```

See Also

Diagram Class
MindFusion.Diagramming Namespace

Diagram Methods

Diagram.addItem Method

Adds an item to the Diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function addItem (item)
```

Parameters

item
The DiagramItem to add.

Remarks

This method is called internally by methods of the Factory class. You should call it explicitly only if instantiating items using the **new** operator.

Example

JavaScript Copy Code

```
var node1 = new ShapeNode(diagram);
node1.bounds = new Drawing.Rect(20, 20, 30, 30);
diagram.addItem(node1);

var node2 = diagram.factory.createShapeNode(60, 20, 30, 30);
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
items Property
nodes Property
links Property
validityChecks Property

Diagram.alignPointToGrid Method

Returns a point of the alignment grid nearest to the one passed as an argument.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function alignPointToGrid (point)
```

Parameters

point
Point. A point specified in diagram coordinates.

Return Value

Point. A point from the alignment grid.

Remarks

The control aligns items to the alignment grid only during user interaction. This method lets you align coordinates of programmatically created or modified items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
showGrid Property
alignToGrid Property

Diagram.arrange Method

Arranges the diagram with the given layout.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function arrange (layout)
```

Parameters

layout

Object. The Layout algorithm to apply.

Remarks

Available layout algorithms are SpringLayout, TreeLayout, BorderedTreeLayout, LayeredLayout, TreeMapLayout, FractalLayout, FlowchartLayout, TopologicalLayout, OrthogonalLayout, RadialTreeLayout, GridLayout.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.arrangeAnimated Method

Arranges the diagram using the given layout, and animates items moving to their new positions.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function arrangeAnimated (layout, [duration, [animationType, [easingType]]])
```

Parameters

layout
Object. The Layout algorithm to apply.

duration
Optional.
Number. An integer, specifying the duration of the animation in milliseconds.

animationType
Optional.
A member of the AnimationType enumeration, specifying the animation type to use.

easingType
Optional.
A member of the EasingType enumeration, specifying the type of easing function to apply to the animation.

Remarks

Available layout algorithms are SpringLayout, TreeLayout, BorderedTreeLayout, LayeredLayout, TreeMapLayout, FractalLayout, FlowchartLayout, TopologicalLayout, OrthogonalLayout, RadialTreeLayout, GridLayout.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.arrangeLinkLabels Method

Rearranges link labels whose autoArrange property is enabled.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function arrangeLinkLabels ()
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LinkLabel Class

Diagram.cancelCompositeOperation Method

Stops recording undo/redo commands into a CompositeCommand and discards the records created so far.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function cancelCompositeOperation ()
```

Remarks

You must call either commitCompositeOperation or this method when the diagram modifications are done after a startCompositeOperation call.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
undoEnabled Property

Diagram.clearAll Method

Removes all items from the diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function clearAll ()
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
addItem Method
items Property

Diagram.commitCompositeOperation Method

Stops recording undo/redo commands into a CompositeCommand and saves the composite in the history queue.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function commitCompositeOperation ()
```

Remarks

You must call either cancelCompositeOperation or this method when the diagram modifications are done after a startCompositeOperation call.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
undoEnabled Property

Diagram.executeCommand Method

Executes the specified command on this diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function executeCommand (command)
```

Parameters

command
The Command to execute.

Remarks

If undo/redo is enabled, the command is saved in the undo history or added to the active composite command.

AddItemCommand and RemoveItemCommand objects are created automatically by the control when adding or removing items, both programmatically or interactively. Calling this method explicitly could be necessary in order to save item's state for later undo using a ChangeItemCommand, or to run a custom operation represented by a class derived from Command.

Example

The following example shows the pattern that should be used to save the state of an item for later undo.

JavaScript Copy Code

```
var ChangeItemCommand = MindFusion.Diagramming.ChangeItemCommand;
var DiagramNode = MindFusion.Diagramming.DiagramNode;

var diagram = DiagramView.find("diagram").diagram;
var node = diagram.activeItem;
if (node instanceof DiagramNode)
{
    var change = new ChangeItemCommand(diagram, node);
    node.text = "new text";
    node.brush = "red";
    diagram.executeCommand(change);
}
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
undoEnabled Property
undo Method
startCompositeOperation Method

Diagram.fromJson Method

Deserializes the diagram from a JSON string.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function fromJson (json)
```

Parameters

json
A string created by the toJson method.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.fromXmlDocument Method

Loads diagram contents from specified XML Document.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function fromXmlDocument (document)
```

Parameters

document
Document. A DOM Document object containing XML-serialized diagram contents.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
toXmlDocument Method

Diagram.getContentBounds Method

Returns the smallest rectangle that bounds all diagram items.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function getContentBounds ([onlyVisible, [repaintBounds]])
```

Parameters

onlyVisible
Optional.
Boolean. Specifies whether only visible items should be considered.

repaintBounds
Optional.
Boolean. Specifies whether to consider the items' repaint bounds.

Return Value

Rect. A Rect instance specifying the smallest rectangle that bounds all diagram items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.getItemAt Method

Finds and returns the topmost DiagramItem found at the specified location.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function getItemAt (point, [onlyInteractive])
```

Parameters

point
A Point instance identifying a diagram location.

onlyInteractive
Optional.
true to return only items that can be interacted with; otherwise, **false**.

Return Value

The topmost DiagramItem found at the given location or **null** if no **DiagramItem** has been found.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.getItemsAt Method

Finds and returns an array of the diagram items found at the specified location.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function getItemsAt (point)
```

Parameters

point
Point. instance identifying a location in the diagram.

Return Value

Array of the DiagramItems found at the given location or **null** if no items have been found. The items are arranged according to their Z-index.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.getLinkAt Method

Finds and returns the topmost DiagramLink found at the specified location.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function getLinkAt (point)
```

Parameters

point
A Point instance identifying a diagram location.

Return Value

The topmost DiagramLink found at the given location or **null** if no **DiagramLink** has been found.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.getLinksAt Method

Finds and returns an array of the diagram DiagramLinks found at the specified location.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function getLinksAt (point)
```

Parameters

point
Point. instance identifying a diagram location.

Return Value

Array of the DiagramLinks found at the given location or **null** if no **DiagramLink**s have been found. The links are arranged according to their Z-index.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.getNearestNode Method

Returns the node nearest to the specified point.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function getNearestNode (point, maxDistance, ignored)
```

Parameters

point
Point. A point in the diagram.

maxDistance
Number. The maximum distance from the point at which to look for nodes.

ignored
DiagramNode. A node that should be ignored.

Return Value

The DiagramNode nearest to the specified point.

Remarks

The last argument is useful when calling this method from nodeCreated and nodeModified handlers and the affected node should not be included in the search.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.getNodeAt Method

Finds and returns the topmost DiagramNode found at the specified location.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function getNodeAt (point, [onlyInteractive])
```

Parameters

point
A Point instance identifying a diagram location.

onlyInteractive
Optional.
true to return only items that can be interacted with; otherwise, **false**.

Return Value

The topmost DiagramNode found at the given location or **null** if no **DiagramNode** has been found.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.getNodesAt Method

Finds and returns an array of the DiagramNodes found at the specified location.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function getNodesAt (point)
```

Parameters

point
Point. instance identifying a diagram location.

Return Value

Array of the DiagramNodes found at the given location or null if no **DiagramNode**s have been found. The nodes are arranged according to their Z-index.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.isItemInteractive Method

Checks if the specified item can be modified interactively.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function isItemInteractive (item)
```

Parameters

item
DiagramItem. The DiagramItem to check.

Return Value

Boolean. **true**, if the item is currently visible and unlocked and it is not associated with a hidden or locked layer, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.isItemVisible Method

Checks if the specified item is currently visible.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function isItemVisible (item)
```

Parameters

item
DiagramItem. The DiagramItem to check.

Return Value

Boolean. **true** if the item's visible property is set to **true** and the item is not associated with a hidden layer, or **false** otherwise.

See Also

Diagram.loadFromString Method

Loads diagram contents from specified XML or JSON string.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function loadFromString (str)
```

Parameters

str
String. A string containing the serialized diagram contents.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
saveToString Method

Diagram.loadFromXml Method

Loads the diagram from an XML file.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function loadFromXml (fileUrl, [onLoad, [onError]])
```

Parameters

fileUrl
String. The URL of an XML file where the data should be read from.

onLoad
Optional.
function. A callback that should be invoked if the file is loaded successfully.

onError
Optional.
function. A callback that should be invoked if the file could not be downloaded.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.raiseClicked Method

Raises an event which notifies that a Diagram item has been clicked.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function raiseClicked (item, mousePosition, mouseButton)
```

Parameters

item
DiagramItem. The Diagram item.

mousePosition
Point. The position of the mouse.

mouseButton
MouseButton. The mouse button, with which the item has been clicked.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.raiseCreated Method

Raises an event which notifies that a Diagram item has been created.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function raiseCreated (item)
```

Parameters

item
DiagramNode | DiagramLink. The diagram item that has been created.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.raiseDeleted Method

Raises an event which notifies that a Diagram item has been deleted.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function raiseDeleted (item)
```

Parameters

item
DiagramItem. The DiagramItem that has been deleted.

See Also

Diagram Members
Diagram Class

Diagram.raiseDeselected Method

Raises an event which notifies that a Diagram item has been deselected.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function raiseDeselected (item)
```

Parameters

item
DiagramItem. The Diagram item.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.raiseDoubleClicked Method

Raises an event which notifies that a Diagram item has been double-clicked.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function raiseDoubleClicked (item, mousePosition, mouseButton)
```

Parameters

item
DiagramItem. The Diagram item.

mousePosition
Point. The position of the mouse.

mouseButton
MouseButton. The mouse button, with which the item has been clicked.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.raiseModified Method

Raises an event which notifies that a Diagram item has been modified.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function raiseModified (item, mousePosition, adjustmentHandle)
```

Parameters

item
DiagramItem. The DiagramItem that has been modified.

mousePosition
Point. The current mouse pointer position.

adjustmentHandle
Number. Identifies the handle used to modify the item.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.raisePasted Method

Raises an event which notifies that a Diagram item has been pasted from clipboard.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function raisePasted (item)
```

Parameters

item
DiagramItem. The DiagramItem that has been pasted.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.raiseSelected Method

Raises an event which notifies that a Diagram item has been selected.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function raiseSelected (item)
```

Parameters

item
DiagramItem. The Diagram item.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.raiseSelecting Method

Raises an event which notifies that a Diagram item will be selected.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function raiseSelecting (item)
```

Parameters

item
DiagramItem. The Diagram item.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.raiseSelectionChanged Method

Raises an event which notifies that the selection has changed.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function raiseSelectionChanged (oldItems, newItems)
```

Parameters

oldItems
Array. The selected items before the change.

newItems
Array. The selected items after the change.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.redo Method

Executes again an action saved in the command history.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function redo ()
```

Remarks

Calling this method has no effect before undo has been called at least once. Any command that has been undone can later be redone by **redo**. The method processes the commands in the same order they were initially executed, which is opposite to the

order in which they are undone.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
undoEnabled Property
executeCommand Method
actionRedone Event

Diagram.registerClass Method

Registers a diagram item class.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function registerClass (itemClass, className, [classId, [classVersion, [baseType]]])
```

Parameters

itemClass
Object. An object identifying the item class.

className
String. A class identifier to use when saving and loading items of the specified type to/from JSON.

classId
Optional.
String. A class identifier to use when saving and loading items of the specified type to/from XML.

classVersion
Optional.
Number. A revision number of the item's class serialization format.

baseType
Optional.
Object. The base class of the item class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.registerComponent Method

Registers a component class.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function registerComponent (componentClass, className)
```

Parameters

componentClass
Object. The component class.

className
String. The name that refers to the component class in CompositeNode templates.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.registerItemClass Method

Registers a diagram item class for XML serialization support.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function registerItemClass (itemClass, classId, classVersion)
```

Parameters

itemClass
Object. An object identifying the item class.

classId
String. A class identifier to use when saving and loading items of the specified type.

classVersion
Number. A revision number of the item's class serialization format.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.removeItem Method

Removes the specified item from the Diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function removeItem (item)
```

Parameters

item
The DiagramItem to be removed.

Remarks

The Diagram object calls this method automatically when users delete items. Classes that inherit **Diagram** can override it to implement some additional processing, but in any case they must call the base method.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
addItem Method
delKeyAction Property
showDeleteButton Property

Diagram.resizeToFitItem Method

Resize the document extents so that the specified item is contained within.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function resizeToFitItem (item)
```

Parameters

item
DiagramItem. The item to be contained.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
bounds Property

Diagram.resizeToFitItems Method

Resizes the document scrollable area so that it fits all diagram items.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function resizeToFitItems ([margin, [onlyVisible, [repaintBounds]]])
```

Parameters

margin
Optional.

onlyVisible
Optional.
Boolean. Specifies whether only visible items should be considered.

repaintBounds
Optional.
Boolean. Specifies whether to consider the items' repaint bounds.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
bounds Property

Diagram.routeAllLinks Method

Routes all links in the diagram in such a way they do not cross nodes.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function routeAllLinks ()
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
autoRoute Property
routeLinks Property
obstacle Property
linkRouted Event

Diagram.saveToString Method

Encodes the diagram contents into an XML or JSON string.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function saveToString (format)
```

Parameters

format
A member of the SaveToStringFormat enumeration.

Return Value

String. A string containing the diagram's serialized representation.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.saveToXml Method

Saves the diagram to an XML file.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function saveToXml (url)
```

Parameters

url
String. A URL specifying where the diagram's XML should be posted to.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
loadFromXml Method

Diagram.startCompositeOperation Method

Starts recording undo/redo commands into a CompositeCommand.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function startCompositeOperation ()
```

Remarks

After the diagram modifications are done, call commitCompositeOperation or cancelCompositeOperation.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
undoEnabled Property

Diagram.toJson Method

Serializes the diagram into a JSON string.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function toJson ()
```

Return Value

String. A string containing the diagram's JSON representation.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
fromJson Method

Diagram.toXmlDocument Method

Saves the diagram into an XML Document.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
function toXmlDocument ()
```

Return Value

Document. A DOM Document object containing XML-serialized diagram contents.

See Also

Diagram.undo Method

Undoes an action saved in the command history.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function undo ()
```

Remarks

The control supports unlimited undo/redo depth. This method moves the "current command" pointer one position back, and calling it again will undo the previous command. The commands can later be redone by calling redo.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
undoEnabled Property
actionUndone Event

Diagram Properties

Diagram.activeItem Property

Gets the active diagram item.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get activeItem() {}
```

Property Value

DiagramItem

The active item, or **null** if no item is active.

Remarks

activeItem is set to the last selected item, and has a distinct appearance from other items as set in activeItemHandlesStyle. Some possible uses for it are as target position when aligning multiple selected items, or as a source of property values when copying styles.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
selection Property

Diagram.activeItemHandlesStyle Property

Gets or sets the visual style of adjustment handles of the active item.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
get activeItemHandlesStyle() {}
set activeItemHandlesStyle(value) {}
```

Property Value

HandlesVisualStyle

A HandlesVisualStyle instance describing handles' appearance.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
activeItem Property
selectedItemHandlesStyle Property
disabledHandlesStyle Property

Diagram.adjustmentHandlesSize Property

Gets or sets the size of adjustment handles that appear around selected items.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
get adjustmentHandlesSize() {}
set adjustmentHandlesSize(value) {}
```

Property Value

Number

A number specifying the size of selection handles. The default value is 2.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.alignToGrid Property

Gets or sets a value indicating whether the diagram items should be aligned to a grid.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
get alignToGrid() {}
set alignToGrid(value) {}
```

Property Value

Boolean

true if the diagram items should be aligned to a grid, or **false** otherwise.

Remarks

When the property is set to true, the sides of every newly created diagram item are aligned to the nearest grid points.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
showGrid Property
gridStyle Property
gridSizeX Property
gridSizeX Property

Diagram.allowLinksRepeat Property

Gets or sets a value indicating whether more than one links can connect the same origin and destination diagram nodes.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get allowLinksRepeat() {}
set allowLinksRepeat(value) {}
```

Property Value

Boolean

true to let links connect to the same nodes, or **false** otherwise. The default is **true**.

Remarks

Specifies whether more than one link can connect the same origin and destination diagram nodes. If set to **false**, two diagram nodes can be connected with at most one link in a single direction.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
allowSelfLoops Property
allowUnconnectedLinks Property

Diagram.allowMultipleResize Property

Gets or sets a value indicating whether multiple selected nodes can be resized simultaneously.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get allowMultipleResize() {}
set allowMultipleResize(value) {}
```

Property Value

Boolean

true to let the user resize multiple selected nodes, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.allowSelfLoops Property

Gets or sets a value indicating whether users are allowed to draw reflexive links.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get allowSelfLoops() {}
set allowSelfLoops(value) {}
```

Property Value

Boolean

true to allow reflexive links, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
linkCreating Event
allowUnconnectedLinks Property
allowLinksRepeat Property

Diagram.allowSplitLinks Property

Gets or sets a value indicating whether link segments can be added and removed interactively.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get allowSplitLinks() {}
set allowSplitLinks(value) {}
```

Property Value

Boolean

true if interactive adding/removing of link segments is allowed, or **false** otherwise.

Remarks

You can insert new link segments interactively by clicking and dragging a point of a selected link's segment. For polyline-style links this creates one additional segment by splitting the existing one into two parts. For cascading links, there are two segments inserted, keeping intact the alternating pattern of horizontal and vertical segments, each orthogonal to its neighbors.

Segments can be merged by moving their joining point until the angle between reaches certain threshold value. In case of cascading and Bezier links, segments can be merged by placing adjacent control points close to each other, as set in mergeThreshold.

This property also lets users change the orientation of cascading links' end segments, by dragging second or next-to-last control point diagonally.

See Also

Diagram.allowUnanchoredLinks Property

Gets or sets a value indicating whether users can link nodes that do not have designated anchor points.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get allowUnanchoredLinks() {}
set allowUnanchoredLinks(value) {}
```

Property Value

Boolean

true to let links connect to nodes without anchor points, or **false** otherwise. The default is **true**.

Remarks

allowUnanchoredLinks applies to nodes whose anchorPattern is not set or does not contain incoming or outgoing anchor points available. If the property is enabled, users can connect links to any point of such nodes. If disabled, users cannot connect links to such nodes.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AnchorPoint Class
AnchorPattern Class

Diagram.allowUnconnectedLinks Property

Gets or sets a value indicating whether users are allowed to draw links that are not connected to any node.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get allowUnconnectedLinks() {}
set allowUnconnectedLinks(value) {}
```

Property Value

Boolean

true to let users draw unconnected links, or **false** otherwise.

Remarks

If this property is set to **true**, users will be able to draw links from or to any point of the document. To draw links that do not start from a node, you must also set behavior to DrawLinks.

Depending on which end of a link is unconnected, its origin or destination nodes will be set to a DummyNode instance.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
linkCreating Event

allowSelfLoops Property
allowLinksRepeat Property

Diagram.autoArrangeAvoidContainers Property

Gets or sets a value indicating whether auto-arranged link labels should avoid containers.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get autoArrangeAvoidContainers() {}
set autoArrangeAvoidContainers(value) {}
```

Property Value

Boolean

true to avoid containers, or **false** otherwise. The default value is **true**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.autoArrangeAvoidSegments Property

Gets or sets a value indicating whether auto-arranged link labels should avoid link segments.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get autoArrangeAvoidSegments() {}
set autoArrangeAvoidSegments(value) {}
```

Property Value

Boolean

true to avoid link segments, or **false** otherwise. The default value is **true**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.autoCloseDistance Property

Gets or sets the maximum distance between first and last points of a FreeFormNode for which the node's outline is closed automatically.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get autoCloseDistance() {}
set autoCloseDistance(value) {}
```

Property Value

Number

A number specifying auto-close distance.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.autoDeleteLinks Property

Gets or sets whether links are deleted automatically when their origin or destination is deleted.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get autoDeleteLinks() {}
set autoDeleteLinks(value) {}
```

Property Value

Boolean

true to delete links along with the nodes they connect, or **false** otherwise. The default value is **true**.

Remarks

If this property is set to **false**, deleting a node will leave its links in the diagram as unconnected links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
delKeyAction Property

Diagram.autoResize Property

Gets or sets a value indicating whether the document area should be resized automatically so it fits the diagram contents.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get autoResize() {}
set autoResize(value) {}
```

Property Value

AutoResize

A member of the AutoResize enumeration indicating the auto resize method. The default is **RightAndDown**.

Remarks

If automatic resizing is enabled, the size of the document scrollable area is updated each time an item is created, deleted or moved by the end-user. To resize the diagram after programmatic changes, call the resizeToFitItems method.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
bounds Property
autoScroll Property

Diagram.backBrush Property

Gets or sets an object specifying how to fill the diagram background.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get backBrush() {}
set backBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.backgroundImageAlign Property

Gets or sets a value indicating how to align the background image.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get backgroundImageAlign() {}
set backgroundImageAlign(value) {}
```

Property Value

ImageAlign

A member of the ImageAlign enumeration.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
backgroundImageUrl Property

Diagram.backgroundImageContent Property

Gets or sets a Base64-encoded image to display as diagram background.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get backgroundImageContent() {}
set backgroundImageContent(value) {}
```

Property Value

String

A Base64-encoded representation of an image.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
backgroundImageAlign Property

Diagram.backgroundImageUrl Property

Gets or sets URL of the image to paint as diagram's background.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get backgroundImageUrl() {}
set backgroundImageUrl(value) {}
```

Property Value

String

A string containing image URL.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
backBrush Property
backgroundImageAlign Property

Diagram.crossingRadius Property

Gets or sets the radius length of decorations displayed at link intersection points.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get crossingRadius() {}
set crossingRadius(value) {}
```

Property Value

Number

A number specifying the radius of the arcs drawn at link intersection points. The default value is 1.5.

Remarks

If linkCrossings is set to Cut, link segments are cut at intersection points and this property defines the length of the break left between adjacent cut points. If **linkCrossings** is set to **Arcs**, this property defines the radius of the arcs drawn over intersection points.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
roundedLinksRadius Property

Diagram.customLinkType Property

Gets or sets the type of DiagramLink objects to create when users starts drawing.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get customLinkType() {}
set customLinkType(value) {}
```

Property Value

Object

An object identifying a constructor function or a class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.customNodeType Property

Gets or sets the type of DiagramNode objects to create when users starts drawing.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get customNodeType() {}
set customNodeType(value) {}
```

Property Value

Object

An object identifying a constructor function or a class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.defaultShape Property

Gets or sets the default shape assigned to new ShapeNode objects.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get defaultShape() {}
set defaultShape(value) {}
```

Property Value

Shape | *String*

A Shape object or identifier.

Remarks

The shape property of newly created ShapeNode objects is initialized with the value of this property. It defines a series of lines, arcs and curves composed in a complex shape. There are predefined shapes exposed as static members of the Shapes class. You can also define your own shapes as shown below.

Example

JavaScript Copy Code

```
var shape = new MindFusion.Diagramming.Shape({
   id: "DoubleEllipse",
   outline: "M100,50 A50,50,0,1,1,0,50 A50,50,0,1,1,100,50",
   decoration: "M98,50 A48,48,0,1,1,2,50 A48,48,0,1,1,98,50",
   fillMode: "nonzero"});

diagram.defaultShape = shape;
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.dirty Property

Gets or sets a value indicating whether the diagram has changed since loading it.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get dirty() {}
set dirty(value) {}
```

Property Value

Boolean

true if the diagram has changed, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.disabledHandlesStyle Property

Gets or sets the visual style of disabled adjustment handles.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get disabledHandlesStyle() {}
set disabledHandlesStyle(value) {}
```

Property Value

HandlesVisualStyle

A HandlesVisualStyle instance describing handles' appearance.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
enabledHandles Property
activeItemHandlesStyle Property
selectedItemHandlesStyle Property

Diagram.dynamicLinks Property

Gets or sets initial value of the dynamic property of new links.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get dynamicLinks() {}
set dynamicLinks(value) {}
```

Property Value

Boolean

true if dynamic links are enabled, or **false** otherwise. The default value is **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.enableSpatialIndex Property

Gets or sets whether to index item positions for faster hit-testing and viewport clipping queries.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get enableSpatialIndex() {}
set enableSpatialIndex(value) {}
```

Property Value

Boolean

true to enable spatial indexing, or **false** otherwise. The default value is **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.expandButtonAction Property

Gets or sets how the control responds to users clickingthe +/- button displayed by expandable nodes.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get expandButtonAction() {}
set expandButtonAction(value) {}
```

Property Value

ExpandButtonAction

A member of the ExpandButtonAction enumeration.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.expandOnIncoming Property

Gets or sets a value indicating the link direction in which tree branches are expanded.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get expandOnIncoming() {}
set expandOnIncoming(value) {}
```

Property Value

Boolean

true if trees are expanded in the direction of incoming links, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.factory Property

Gets a Factory object that lets you programmatically add new items to the diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get factory() {}
```

Property Value

Factory

An instance of the Factory class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.font Property

Gets or sets the default font of text displayed by diagram items.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get font() {}
set font(value) {}
```

Property Value

Font

A font used to render the text of diagram items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.freeFormAttractDistance Property

Gets or sets the radius around dragged free-form adjustment handle in which other points of FreeFormNode are modified too.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get freeFormAttractDistance() {}
set freeFormAttractDistance(value) {}
```

Property Value

Number

A number specifying point attraction radius.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.freeFormTargets Property

Gets or sets an array of Shape objects used to replace FreeFormNode instances with ShapeNode ones when behavior is set to LinkFreeShapes or **DrawFreeShapes**.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get freeFormTargets() {}
set freeFormTargets(value) {}
```

Property Value

Array

An array of shape identifiers or Shape instances.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.gridColor Property

Gets or sets the color of grid points.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get gridColor() {}
set gridColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.gridOffsetX Property

Gets or sets the horizontal offset of the first point of the alignment grid.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get gridOffsetX() {}
set gridOffsetX(value) {}
```

Property Value

Number

The horizontal offset of the first point of the alignment grid.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.gridOffsetY Property

Gets or sets the vertical offset of the first point of the alignment grid.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get gridOffsetY() {}
set gridOffsetY(value) {}
```

Property Value

Number

The vertical offset of the first point of the alignment grid.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.gridPointSize Property

Gets or sets the size of shapes used to represent grid points.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get gridPointSize() {}
set gridPointSize(value) {}
```

Property Value

Number

The size of shapes used to represent grid points.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.gridSizeX Property

Gets or sets the horizontal distance between adjacent grid points.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get gridSizeX() {}
set gridSizeX(value) {}
```

Property Value

Number

The horizontal distance between adjacent grid points.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.gridSizeY Property

Gets or sets the vertical distance between adjacent grid points.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get gridSizeY() {}
set gridSizeY(value) {}
```

Property Value

Number

The vertical distance between adjacent grid points.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.gridStyle Property

Gets or sets the visual style of the alignment grid.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get gridStyle() {}
set gridStyle(value) {}
```

Property Value

GridStyle

A member of the GridStyle enumeration.

See Also

Diagram.items Property

Returns an array of all items in this diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get items() {}
```

Property Value

Array

Array of DiagramItem objects.

Remarks

This collection gives you access to all items contained in this diagram. You cannot add or remove items directly through it, but must use the addItem and removeItem methods instead.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
enableSpatialIndex Property

Diagram.laneGrid Property

Gets the lane grid.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get laneGrid() {}
```

Property Value

Lanes.Grid

An instance of the Lanes.Grid class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.linkBaseShape Property

Gets or sets the shape displayed at the base of new links.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get linkBaseShape() {}
set linkBaseShape(value) {}
```

Property Value

Shape | *String*

A Shape instance or identifier that represents an arrowhead shape.

Remarks

This property specifies the default shape displayed as arrowhead at the links' origin points.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ArrowHeads Class
baseShape Property

Diagram.linkBaseShapeSize Property

Gets or sets the default size of arrowheads at base of new links.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get linkBaseShapeSize() {}
set linkBaseShapeSize(value) {}
```

Property Value

Number

A number specifying arrowhead size. The default value is 5.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.linkBrush Property

Gets or sets the default fill of link arrowheads.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get linkBrush() {}
set linkBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

Remarks

The property provides a default value for links' brush, used to fill the interior of arrowheads.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.linkCrossings Property

Gets or sets a value indicating how to render the intersection points of links.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get linkCrossings() {}
set linkCrossings(value) {}
```

Property Value

LinkCrossings

A member of the LinkCrossings enumeration. The default value is **Straight**.

Remarks

Intersection points where links cross each other's paths can be rendered as arcs or cuts, depending on the value of this property. The diagram z-order defines which links jump over the others or which links are cut by the others. If **linkCrossings** is set to Arcs, links with higher zIndex jump over those with lower **zIndex**. If set to **Cut**, segments of links at lower z-position are cut by links at higher z-position. Radius of arcs and cuts can be set via the crossingRadius property.

Performance Note

Calculating coordinates for crossing point decorations is CPU intensive operation and is triggered each time link's control points change. If populating large diagrams from data, best enable this property only after creating all links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
routeLinks Property
roundedLinks Property
validateLinkCrossing Event

Diagram.linkHeadShape Property

Gets or sets the shape displayed as head of new links.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get linkHeadShape() {}
set linkHeadShape(value) {}
```

Property Value

Shape | String

A Shape instance or identifier that represents an arrowhead shape. The default is Arrow.

Remarks

This property specifies the default shape displayed as arrowhead at the links' end points.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ArrowHeads Class
headShape Property

Diagram.linkHeadShapeSize Property

Gets or sets the default size of arrowheads.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get linkHeadShapeSize() {}
set linkHeadShapeSize(value) {}
```

Property Value

Number

A number specifying arrowhead size. The default value is 5.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.linkHitDistance Property

Gets or sets a value specifying how far from a link a click is still considered a hit.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get linkHitDistance() {}
set linkHitDistance(value) {}
```

Property Value

Number

A value specifying link hit distance.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.linkPen Property

Gets or sets the default stroke of links.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get linkPen() {}
set linkPen(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

Remarks

This property specifies the default value for the pen property of new links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.linkRouter Property

Gets or sets the object used to find paths for auto-routed links when diagram nodes are added or their positions change.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get linkRouter() {}
set linkRouter(value) {}
```

Property Value

Router

An instance of Router-derived class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
linkRouted Event

Diagram.links Property

Returns an array of all links in this diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get links() {}
set links(value) {}
```

Property Value

Array

Array of DiagramLink objects.

Remarks

You cannot add or remove links directly through this array, but must use the addItem and removeItem methods instead.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.linkSegments Property

Gets or sets the default number of link segments.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get linkSegments() {}
set linkSegments(value) {}
```

Property Value

Number

The default number of link segments.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.linkShape Property

Gets or sets the initial shape assigned to new links.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get linkShape() {}
set linkShape(value) {}
```

Property Value

LinkShape

A member of the LinkShape enumeration. The default is **Polyline**.

Remarks

The specified value is assigned to the shape property of new links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
roundedLinks Property

Diagram.linkTextStyle Property

Gets or sets the default text placement and orientation of new links.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
get linkTextStyle() {}
set linkTextStyle(value) {}
```

Property Value

LinkTextStyle

A member of the LinkTextStyle enumeration. The default value is **Center**.

Remarks

The value of this property is used to initialize the textStyle property of new links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.mergeThreshold Property

Gets or sets the maximum distance between adjacent control points of a link at which the respective segments can be merged.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
get mergeThreshold() {}
set mergeThreshold(value) {}
```

Property Value

Number

A number specifying the maximum distance. The default is 0.

Remarks

A user may add or remove links' segments interactively when the allowSplitLinks property is enabled. They can drag a link's control point over the next one in order to merge two adjacent segments. Usually the two points must be placed exactly at the same coordinates in order to merge the respective segments of the link. Assign a larger value to this property if you wish to make it easier for users to merge segments by just moving link's control point close to each other.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.nodeEffects Property

Gets the global effects applied to nodes in this diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get nodeEffects() {}
```

Property Value

Array

An array of NodeEffect objects.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.nodes Property

Returns an array of all nodes in this diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get nodes() {}
```

Property Value

Array

Array of DiagramNode objects.

Remarks

You cannot add or remove nodes directly through this array, but must use the addItem and removeItem methods instead.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.nodesExpandable Property

Gets or sets a value indicating whether newly created nodes are expandable.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get nodesExpandable() {}
set nodesExpandable(value) {}
```

Property Value

Boolean

A boolean value assigned to the expandable property of new nodes.

See Also

Diagram.roundedLinks Property

Gets or sets a value indicating whether the joints between segments of Polyline and **Cascading** links should be rounded.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get roundedLinks() {}
set roundedLinks(value) {}
```

Property Value

Boolean

true to draw rounded segment joints, or **false** otherwise. The default value is **false**.

Remarks

The value of this property is considered when drawing links whose shape is set to Polyline or **Cascading**. If the property is enabled, the joins between segments of all such links are drawn as arcs, whose radius equals the value of roundedLinksRadius.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
linkCrossings Property

Diagram.roundedLinksRadius Property

Gets or sets the radius of joint arcs of rounded links' segments.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get roundedLinksRadius() {}
set roundedLinksRadius(value) {}
```

Property Value

Number

A number specifying the radius length. The default value is 2.

Remarks

The value of this property is used when roundedLinks is enabled. It specifies the radius of the arcs drawn as joins between link segments.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
crossingRadius Property

Diagram.routeLinks Property

Gets or sets initial value of the autoRoute property of new links.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
get routeLinks() {}
set routeLinks(value) {}
```

Property Value

Boolean

true to route links, or **false** otherwise. The default value is **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
linkRouter Property
linkCrossings Property
linkRouted Event

Diagram.routeMargin Property

Gets or sets the minimal distance between nodes and auto-routed links.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
get routeMargin() {}
set routeMargin(value) {}
```

Property Value

Number

A value specifying distance in the current measureUnit. The default value is 8.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.selectedItemHandlesStyle Property

Gets or sets the visual style of adjustment handles of selected items.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
get selectedItemHandlesStyle() {}
set selectedItemHandlesStyle(value) {}
```

Property Value

HandlesVisualStyle

A HandlesVisualStyle instance describing handles' appearance.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
selection Property
activeItemHandlesStyle Property
disabledHandlesStyle Property

Diagram.selection Property

Gets a Selection instance that represents the selection of items in this diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get selection() {}
```

Property Value

Selection

An instance of the Selection class representing the current selection.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
activeItem property
allowMultipleSelection Property

Diagram.shadowOffsetX Property

Gets or sets the horizontal offset of items' shadows.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get shadowOffsetX() {}
set shadowOffsetX(value) {}
```

Property Value

Number

A number specifying shadows' horizontal offset.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.shadowOffsetY Property

Gets or sets the vertical offset of items' shadows.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
get shadowOffsetY() {}
set shadowOffsetY(value) {}
```

Property Value

Number

A number specifying shadows' vertical offset.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.shadowsStyle Property

Gets or sets the style of shadows.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
get shadowsStyle() {}
set shadowsStyle(value) {}
```

Property Value

ShadowsStyle

A member of the ShadowsStyle enumeration.

Remarks

This property specifies how the shadows of diagram items are drawn. If its value is None, there are no shadows painted. **OneLevel** makes all shadows paint on the lowest z-level, not covering any items. Thus the diagram items will seem to appear on the same z-level. **ZOrder** makes the shadow paint immediately under their items, possibly covering other items. This style gives more 3D-like outlook to the diagram.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.shapeBrush Property

Gets or sets the default fill of ShapeNode objects.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript — Copy Code

```
get shapeBrush() {}
set shapeBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

Remarks

This property provides a default value for the brush used to fill ShapeNode objects.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.shapeLibraryLocation Property

Gets or sets the location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get shapeLibraryLocation() {}
set shapeLibraryLocation(value) {}
```

Property Value

String

The location of a shape library file containing custom shape definitions.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.shapePen Property

Gets or sets the default stroke of ShapeNode objects.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get shapePen() {}
set shapePen(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

Remarks

This property provides a default value for the pen used to stroke ShapeNode border lines.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.showAnchors Property

Gets or sets a value indicating when to display anchor point marks.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get showAnchors() {}
set showAnchors(value) {}
```

Property Value

ShowAnchors

A member of the ShowAnchors enumeration.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.showDisabledHandles Property

Gets or sets a value indicating whether adjustment handles are painted when they are disabled.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get showDisabledHandles() {}
set showDisabledHandles(value) {}
```

Property Value

Boolean

true to display disabled handles, or **false** otherwise. The default value is **true**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.showGrid Property

Gets or sets a value indicating whether the alignment grid is visible.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get showGrid() {}
set showGrid(value) {}
```

Property Value

Boolean

true if the alignment grid is visible, or **false** otherwise.

Remarks

If it is visible, the grid is painted as either a matrix of points or crossing horizontal and vertical lines. That depends on the value of gridStyle.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
alignToGrid Property
gridStyle Property
gridSizeX Property
gridSizeY Property

Diagram.showLaneGrid Property

Gets or sets a value indicating whether the lane grid is visible.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get showLaneGrid() {}
set showLaneGrid(value) {}
```

Property Value

Boolean

true if the lane grid is visible, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
laneGrid Property

Diagram.style Property

Gets or sets the style associated with this diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get style() {}
set style(value) {}
```

Property Value

Style

An instance of the Style class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.tag Property

Gets or sets custom data associated with this diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get tag() {}
set tag(value) {}
```

Property Value

Object

An object containing custom data.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.theme Property

Gets or sets the theme associated with this diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get theme() {}
set theme(value) {}
```

Property Value

Theme

An instance of the Theme class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.undoEnabled Property

Gets or sets a value indicating whether saving action records for later undo is enabled.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get undoEnabled() {}
set undoEnabled(value) {}
```

Property Value

Boolean

true if undo is enabled, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.validityChecks Property

Gets or sets a value indicating whether validity checks are performed each time an item is added to the diagram.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get validityChecks() {}
set validityChecks(value) {}
```

Property Value

Boolean

true to check validity when adding items, or **false** otherwise. The default is **true**.

Remarks

If enabled, validity checks are performed each time an item is added to the diagram. That involves enumerating the items collection and can slow up the process considerably for large diagrams. You can disable this property in order to skip the validity checks, however be sure that you don't add an item twice to the diagram and that links are created only between items in the same diagram.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
enableSpatialIndex Property

Diagram.version Property

Gets a string identifying current version of the library.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
get version() {}
set version(value) {}
```

Property Value

String

A string in "major.minor.revision" format.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram Events

DiagramBase.ActionRecorded Event

Raised when an action is recorded in the undo buffer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<UndoEventArgs> ActionRecorded
```

Visual Basic Copy Code

```
Public Event ActionRecorded As EventHandler(Of UndoEventArgs)
```

Event Data

ActionRecorded event handlers receive an argument of type UndoEventArgs. The following **UndoEventArgs** members provide information relevant to the event:

Member name	Description
Command	Refers to the Command instance that identifies the recorded action.

Remarks

This event is raised only for top-level commands. For example, it isn't raised for the constituent actions of a CompositeCmd.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ActionRedone Event
ActionUndone Event

DiagramBase.ActionRecording Event

A validation event raised before an action is recorded in the undo buffer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<UndoValidationEventArgs> ActionRecording
```

Visual Basic Copy Code

```
Public Event ActionRecording As EventHandler(Of UndoValidationEventArgs)
```

Event Data

ActionRecording event handlers receive an argument of type UndoValidationEventArgs. The following **UndoValidationEventArgs** members provide information relevant to the event:

Member name	Description
Command	Refers to the Command instance that identifies the action being recorded.
Cancel	Set this property to **true** to prevent the action from being recorded in the undo buffer.

Remarks

Do not filter commands that create or delete items and groups. Otherwise, redoing or undoing recorded commands related to the items whose create/delete command is filtered will cause various exceptions. We recommend to filter commands that modify an item, change its selection state, scroll a table, and so on.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ActionRecorded Event
ActionRedone Event

DiagramBase.ActionRedone Event

Raised when an action is redone.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<UndoEventArgs> ActionRedone
```

Visual Basic Copy Code

```
Public Event ActionRedone As EventHandler(Of UndoEventArgs)
```

Event Data

ActionRedone event handlers receive an argument of type UndoEventArgs. The following **UndoEventArgs** members provide information relevant to the event:

Member name	Description
Command	Refers to the Command instance that identifies the recorded action.

Remarks

This event is raised during calls to Redo. It is raised both for top-level commands and for the constituent actions of composite commands.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

ActionRecorded Event
ActionUndone Event

DiagramBase.ActionUndone Event

Raised when an action is undone.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<UndoEventArgs> ActionUndone
```

Visual Basic Copy Code

```
Public Event ActionUndone As EventHandler(Of UndoEventArgs)
```

Event Data

ActionUndone event handlers receive an argument of type UndoEventArgs. The following **UndoEventArgs** members provide information relevant to the event:

Member name	Description
Command	Refers to the Command instance that identifies the recorded action.

Remarks

This event is raised during calls to Undo. It is raised both for top-level commands and for the constituent actions of composite commands.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ActionRecorded Event
ActionRedone Event

DiagramBase.ActiveLayerChanged Event

Raised when the diagram's active layer has changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler ActiveLayerChanged
```

Visual Basic Copy Code

```
Public Event ActiveLayerChanged As EventHandler
```

Event Data

ActiveLayerChanged event handlers receive an argument of type **EventArgs**.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.BeginLoad Event

Raised when the component starts loading diagram data from a file or memory stream.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler BeginLoad
```

Visual Basic Copy Code

```
Public Event BeginLoad As EventHandler
```

Event Data

BeginLoad event handlers receive an argument of type **EventArgs**.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
EndLoad Event

DiagramBase.BoundsChanged Event

Raised when the Bounds property value has changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler BoundsChanged
```

Visual Basic Copy Code

```
Public Event BoundsChanged As EventHandler
```

Event Data

BoundsChanged event handlers receive an argument of type **EventArgs**.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
MeasureUnitChanged Event
ActionRedone Event

DiagramBase.CellClicked Event

Raised when the user clicks a table's cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<CellEventArgs> CellClicked
```

Visual Basic Copy Code

```
Public Event CellClicked As EventHandler(Of CellEventArgs)
```

Event Data

CellClicked event handlers receive an argument of type CellEventArgs. The following **CellEventArgs** members provide information relevant to the event:

Member name	Description
Table	The TableNode that contains the clicked cell.
Cell	The Cell that has been clicked.
MouseButton	The mouse button that has been pressed by the user.
MousePosition	A **PointF** specifying the mouse cursor position.
Column	Index of the column containing the clicked cell.
Row	Index of the row containing the clicked cell.

Remarks

The control also raises NodeClicked event for the **TableNode** after **CellClicked**, unless you set the PreventNodeEvent argument to **true**.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
CellDoubleClicked Event
ActionRedone Event

DiagramBase.CellDoubleClicked Event

Raised when the user double-clicks a table's cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<CellEventArgs> CellDoubleClicked
```

Visual Basic Copy Code

```
Public Event CellDoubleClicked As EventHandler(Of CellEventArgs)
```

Event Data

CellDoubleClicked event handlers receive an argument of type CellEventArgs. The following **CellEventArgs** members provide information relevant to the event:

Member name	Description

Table	The TableNode that contains the clicked cell.
Cell	The Cell that has been double-clicked.
MouseButton	The mouse button that has been pressed by the user.
MousePosition	A **PointF** specifying the mouse cursor position.
Column	Index of the column containing the clicked cell.
Row	Index of the row containing the clicked cell.

Remarks

The control also raises NodeDoubleClicked event for the **TableNode** after **CellDoubleClicked**, unless you set the PreventNodeEvent argument to **true**.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
CellClicked Event

DiagramBase.CellTextEdited Event

Raised when the user edits the text of a table's cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<EditCellTextEventArgs> CellTextEdited
```

Visual Basic Copy Code

```
Public Event CellTextEdited As EventHandler(Of EditCellTextEventArgs)
```

Event Data

CellTextEdited event handlers receive an argument of type EditCellTextEventArgs. The following **EditCellTextEventArgs** members provide information relevant to the event:

Member name	Description
Table	The TableNode that contains the edited cell.
Cell	The Cell whose text has been edited.
NewText	The cell's new text.
OldText	The cell's old text.
Column	Index of the column containing the edited cell.
Row	Index of the row containing the edited cell.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
CellTextEditing Event
ActionRedone Event

DiagramBase.CellTextEditing Event

A validation event that lets you approve or reject changes made to the text of a table's cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<CellValidationEventArgs> CellTextEditing
```

Visual Basic Copy Code

```
Public Event CellTextEditing As EventHandler(Of CellValidationEventArgs)
```

Event Data

CellTextEditing event handlers receive an argument of type CellValidationEventArgs. The following **CellValidationEventArgs** members provide information relevant to the event:

Member name	Description
Table	The TableNode containing the cell that the user is trying to edit.
Cell	The Cell that the user is trying to edit.
Column	Index of the column containing the cell.
Row	Index of the row containing the cell.
Cancel	Set this property to **true** to prevent the user from editing the cell's text.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeTextEditing Event
CellTextEdited Event
ActionRedone Event

DiagramBase.Clicked Event

Raised when the user clicks the diagram at a point where no items are located.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<DiagramEventArgs> Clicked
```

Visual Basic Copy Code

```
Public Event Clicked As EventHandler(Of DiagramEventArgs)
```

Event Data

Clicked event handlers receive an argument of type DiagramEventArgs. The following **DiagramEventArgs** members provide information relevant to the event:

Member name	Description
MouseButton	The mouse button that has been pressed by the user.
MousePosition	A **PointF** specifying the mouse cursor position.

Remarks

This event is raised only if the user clicks an unoccupied point of the diagram. If the user clicks a diagram item, one of the following events is raised: NodeClicked, CellClicked or LinkClicked.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
CellDoubleClicked Event

DiagramBase.ContainerChildAdded Event

Raised when a node is added to a ContainerNode, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<ContainerChildEventArgs> ContainerChildAdded
```

Visual Basic Copy Code

```
Public Event ContainerChildAdded As EventHandler(Of ContainerChildEventArgs)
```

Event Data

ContainerChildAdded event handlers receive an argument of type ContainerChildEventArgs.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ContainerChildRemoved Event

DiagramBase.ContainerChildRemoved Event

Raised when a node is removed from a ContainerNode, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<ContainerChildEventArgs> ContainerChildRemoved
```

Visual Basic Copy Code

```
Public Event ContainerChildRemoved As EventHandler(Of ContainerChildEventArgs)
```

Event Data

ContainerChildRemoved event handlers receive an argument of type ContainerChildEventArgs.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ContainerChildAdded Event

DiagramBase.ContainerFolded Event

Raised when the user folds a container by clicking the arrow-up button in the container's caption bar.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> ContainerFolded
```

Visual Basic Copy Code

```
Public Event ContainerFolded As EventHandler(Of NodeEventArgs)
```

Event Data

ContainerFolded event handlers receive an argument of type NodeEventArgs.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ContainerUnfolded Event
ContainerNode.Foldable Property

DiagramBase.ContainerPageChanged Event

Raised when the user changes current page of a PagedContainerNode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> ContainerPageChanged
```

Visual Basic Copy Code

```
Public Event ContainerPageChanged As EventHandler(Of NodeEventArgs)
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.ContainerUnfolded Event

Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> ContainerUnfolded
```

Visual Basic Copy Code

```
Public Event ContainerUnfolded As EventHandler(Of NodeEventArgs)
```

Event Data

ContainerUnfolded event handlers receive an argument of type NodeEventArgs.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ContainerFolded Event
ContainerNode.Foldable Property

DiagramBase.ContentVisibilityChanged Event

Raised when the visibility of container's content changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> ContentVisibilityChanged
```

Visual Basic Copy Code

```
Public Event ContentVisibilityChanged As EventHandler(Of NodeEventArgs)
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.DefaultShapeChanged Event

Raised when the DefaultShape property value has changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler DefaultShapeChanged
```

Visual Basic Copy Code

```
Public Event DefaultShapeChanged As EventHandler
```

Event Data

DefaultShapeChanged event handlers receive an argument of type **EventArgs**.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.DeserializeTag Event

Raised when the Tag of a diagram item must be deserialized from XML format.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<SerializeTagEventArgs> DeserializeTag
```

Visual Basic Copy Code

```
Public Event DeserializeTag As EventHandler(Of SerializeTagEventArgs)
```

Event Data

DeserializeTag event handlers receive an argument of type SerializeTagEventArgs. The following **SerializeTagEventArgs** members provide information relevant to the event:

Member name	Description
Object	The object whose Tag is being deserialized. This object can be either a diagram item, group, cell or anchor point.
Tag	Set this property to the object that has been deserialized.
Representation	An **XmlElement** instance that contains the XML representation of a Tag value.
Context	An XmlPersistContext object providing information about the document being deserialized, along with methods that help you deserialize data from XML.
Handled	Set this flag to **true** if you perform custom tag deserialization.

Remarks

Only tags of simple value types can be saved and loaded automatically to/from the XML document. You must provide your own serialization for complex types by implementing this event and SerializeTag.

Example

The following example demonstrates how to deserialize a complex tag value using the **DeserializeTag** event.

C# Copy Code

```
private void Read(string fileName)
{
    diagram.DeserializeTag += new SerializeTagEventHandler(DeserializeTag);
    diagram.LoadFromXml(fileName);
}

private void DeserializeTag(object sender, SerializeTagEventArgs e)
{
    e.Handled = true;
    string type, val;
```

```
    XmlText textNode = e.Representation.LastChild as XmlText;
    string stringRep = textNode.Value;
    int pos = stringRep.IndexOf(";");

    if (pos == -1)
        return;

    type = stringRep.Substring(0, pos);
    val = stringRep.Substring(pos + 1);

    Type t = Type.GetType(type);

    object tag = null;
    if (t == typeof(string))
        tag = val;
    else if (t == typeof(int))
        tag = XmlConvert.ToInt32(val);

    e.Tag = tag;
}
```

Visual Basic Copy Code

```
Private Sub Read(ByVal fileName As String)

    AddHandler diagram.DeserializeTag, New SerializeTagEventHandler(AddressOf DeserializeTag)
    diagram.LoadFromXml(fileName)

End Sub

Private Sub DeserializeTag(ByVal sender As Object, ByVal e As SerializeTagEventArgs)

    e.Handled = True

    Dim type As String
    Dim val As String

    Dim textNode As XmlText = CType(e.Representation.LastChild, XmlText)
    Dim stringRep As String = textNode.Value
    Dim pos As Integer = stringRep.IndexOf(";")

    If pos = -1 Then
        Return
    End If

    type = stringRep.Substring(0, pos)
    val = stringRep.Substring(pos + 1)

    Dim t As Type = System.Type.GetType(type)

    Dim tag As Object = Nothing
    If t Is GetType(String) Then
        tag = val
    ElseIf t Is GetType(Integer) Then
        tag = XmlConvert.ToInt32(val)
    End If

    e.Tag = tag

End Sub
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
SerializeTag Event
Diagram.LoadFromXml Method

DiagramBase.DestinationAnchorChanged Event

Raised when the DestinationAnchor property value of a link has changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<AnchorPointEventArgs> DestinationAnchorChanged
```

Visual Basic Copy Code

```
Public Event DestinationAnchorChanged As EventHandler(Of AnchorPointEventArgs)
```

Event Data

DestinationAnchorChanged event handlers receive an argument of type AnchorPointEventArgs. The following **AnchorPointEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink whose DestinationAnchor has changed.
OldAnchorIndex	Index of the old anchor point.
NewAnchorIndex	Index of the new anchor point.

Remarks

This event is raised every time the **DestinationAnchor** property of a link changes, either because a user connects the link end to a different anchor point, or because an application sets it programmatically.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
OriginAnchorChanged Event
ActionRedone Event

DiagramBase.DirtyChanged Event

Raised when the **Dirty** property value has changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler DirtyChanged
```

Visual Basic Copy Code

```
Public Event DirtyChanged As EventHandler
```

Event Data

DirtyChanged event handlers receive an argument of type **EventArgs**.

Remarks

Dirty is set to **true** every time the diagram or the diagram document is modified, but this event is raised only if the flag value changes, either from **false** to **true** or from **true** to **false**. So the event occurs only when a diagram or a diagram document is

modified for the first time, but not when subsequent modifications are done.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ActionRecorded Event

DiagramBase.DoubleClicked Event

Raised when the user double-clicks the diagram at a point where no items are located.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<DiagramEventArgs> DoubleClicked
```

Visual Basic Copy Code

```
Public Event DoubleClicked As EventHandler(Of DiagramEventArgs)
```

Event Data

DoubleClicked event handlers receive an argument of type DiagramEventArgs. The following **DiagramEventArgs** members provide information relevant to the event:

Member name	Description
MouseButton	The mouse button that has been pressed by the user.
MousePosition	A **PointF** specifying the mouse cursor position.

Remarks

This event is raised only if the user double-clicks an unoccupied point of the diagram. If the user double-clicks a diagram item, one of the following events is raised: NodeDoubleClicked, CellDoubleClicked or LinkDoubleClicked.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ActionRedone Event
Clicked Event

DiagramBase.DrawAdjustmentHandles Event

Raised when item's adjustment handles must be custom-drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<DrawItemEventArgs> DrawAdjustmentHandles
```

Visual Basic Copy Code

```
Public Event DrawAdjustmentHandles As EventHandler(Of DrawItemEventArgs)
```

Event Data

DrawAdjustmentHandles event handlers receive an argument of type DrawItemEventArgs. The following **DrawItemEventArgs** members provide information relevant to the event:

Member name	Description
Item	The DiagramItem whose adjustment handles should be drawn.
Graphics	An object implementing the IGraphics interface whose methods should be used to draw the item's handles.

Remarks

DrawAdjustmentHandles is raised for selected items whose HandlesStyle is set to Custom.

Example

The following example illustrates how to draw a red frame around a node as a visualization of its adjustment handles.

C# Copy Code

```
void OnDrawAdjustmentHandles(object source, DrawItemEventArgs e)
{
    var pen = new MindFusion.Drawing.Pen(Colors.Red, 0);

    if (e.Item is ShapeNode shapeNode)
    {
        e.Graphics.DrawRectangle(
            pen, shapeNode.Bounds);
    }

    if (e.Item is DiagramLink link)
    {
        e.Graphics.DrawLines(
            pen, link.ControlPoints);
    }
}
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
HitTestAdjustmentHandles Event

DiagramBase.DrawAnchorPoint Event

Raised when an anchor point must be custom-drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<DrawAnchorPointEventArgs> DrawAnchorPoint
```

Visual Basic Copy Code

```
Public Event DrawAnchorPoint As EventHandler(Of DrawAnchorPointEventArgs)
```

Event Data

DrawAnchorPoint event handlers receive an argument of type DrawAnchorPointEventArgs. The following **DrawAnchorPointEventArgs** members provide information relevant to the event:

Member name	Description
Node	A DiagramNode containing the anchor point that is being painted.
AnchorIndex	Index of the anchor point within the node's **AnchorPattern**.
AnchorPattern	An AnchorPattern instance that contains the anchor point definition.
Location	A **Point** specifying the location of the anchor point.
Graphics	An object implementing the IGraphics interface whose methods should be used to draw the anchor point mark.

Remarks

DrawAnchorPoint is raised for an AnchorPoint if the point's MarkStyle is set to Custom.

Example

The following example demonstrates how to draw images for anchor points:

C# Copy Code

```
void diagram_DrawAnchorPoint(object sender, DrawAnchorPointEventArgs e)
{
    var imgPos = e.Location;
    imgPos.Y -= 1.8;

    e.Graphics.DrawImage(icon, imgPos);
}
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.DrawBackground Event

Raised to let you custom draw the diagram background.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<DiagramEventArgs> DrawBackground
```

Visual Basic Copy Code

```
Public Event DrawBackground As EventHandler(Of DiagramEventArgs)
```

Event Data

DrawBackground event handlers receive an argument of type DiagramEventArgs. The following **DiagramEventArgs** members provide information relevant to the event:

Member name	Description
Graphics	Gets an object implementing the IGraphics interface whose methods should be used to draw the custom graphics.
VisibleRect	Specifies what part of the diagram is currently visible.

ClipRect	Gets the current clipping rectangle.

Remarks

This event is raised just after drawing the BackgroundImage and before drawing the alignment grid.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DrawForeground Event

DiagramBase.DrawBranchIndicator Event

Raised when a branch indicator must be custom-drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<DrawBranchIndicatorEventArgs> DrawBranchIndicator
```

Visual Basic Copy Code

```
Public Event DrawBranchIndicator As EventHandler(Of DrawBranchIndicatorEventArgs)
```

Event Data

Member name	Description
From	A Vector specifying the direction of the incoming link segment.
Graphics	Gets an IGraphics object whose methods should be used to draw the branch indicator.
Merge	**true** to draw a merge indicator, or **false** to draw a branch indicator.
Position	A **PointF** specifying the position of the branch indicator.
To	A Vector specifying the direction of the outgoing link segment.

Remarks

This event is raised when LinkBranchIndicator and/or LinkMergeIndicator properties are set to Custom.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.DrawCell Event

Raised when a table's cell must be custom drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<DrawCellEventArgs> DrawCell
```

Visual Basic Copy Code

```
Public Event DrawCell As EventHandler(Of DrawCellEventArgs)
```

Event Data

DrawCell event handlers receive an argument of type DrawCellEventArgs. The following **DrawCellEventArgs** members provide information relevant to the event:

Member name	Description
Table	The TableNode containing the cell that should be custom drawn.
Cell	The Cell that should be custom drawn.
Column	Index of the column containing the cell.
Row	Index of the row containing the cell.
Bounds	Gets the bounding rectangle of the cell being drawn.
Graphics	An object implementing the IGraphics interface whose methods should be used to draw the cell contents.

Remarks

This event is raised for tables whose CellCustomDraw property is enabled.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DrawNode Event
ActionRedone Event

DiagramBase.DrawForeground Event

Raised to let you custom draw the diagram foreground.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<DiagramEventArgs> DrawForeground
```

Visual Basic Copy Code

```
Public Event DrawForeground As EventHandler(Of DiagramEventArgs)
```

Event Data

DrawForeground event handlers receive an argument of type DiagramEventArgs. The following **DiagramEventArgs** members provide information relevant to the event:

Member name	Description

Graphics	Gets an object implementing the IGraphics interface whose methods should be used to draw the custom graphics.
VisibleRect	Specifies what part of the diagram is currently visible.
ClipRect	Gets the current clipping rectangle.

Remarks

This event is raised after all diagram elements has been rendered, but before drawing the diagram boundaries with BoundsPen.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DrawBackground Event

DiagramBase.DrawLink Event

Raised when a diagram link must be custom drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<DrawLinkEventArgs> DrawLink
```

Visual Basic Copy Code

```
Public Event DrawLink As EventHandler(Of DrawLinkEventArgs)
```

Event Data

DrawLink event handlers receive an argument of type DrawLinkEventArgs. The following **DrawLinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that should be custom drawn.
Points	Gets a collection of the link's control points. When drawing shadows, the points are offset as necessary.
Shadow	**true** if now drawing the link's shadow, otherwise **false**.
Graphics	An object implementing the IGraphics interface whose methods should be used to draw the link.

Remarks

This event is raised for links whose CustomDraw property is enabled.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DrawNode Event

DiagramBase.DrawNode Event

Raised when a diagram node must be custom drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<DrawNodeEventArgs> DrawNode
```

Visual Basic Copy Code

```
Public Event DrawNode As EventHandler(Of DrawNodeEventArgs)
```

Event Data

DrawNode event handlers receive an argument of type DrawNodeEventArgs. The following **DrawNodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that should be custom drawn.
Bounds	The bounding rectangle of the node being drawn.
Shadow	**true** if now drawing the node's shadow, otherwise **false**.
Graphics	An object implementing the IGraphics interface whose methods should be used to draw the node.

Remarks

This event is raised for nodes whose **CustomDraw** property is enabled.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DrawLink Event

DiagramBase.EndLoad Event

Raised when the component ends loading diagram data from a file or memory stream.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler EndLoad
```

Visual Basic Copy Code

```
Public Event EndLoad As EventHandler
```

Event Data

EndLoad event handlers receive an argument of type **EventArgs**.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.ExpandButtonClicked Event

Raised when the user clicks the [±] button of an Expandable node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> ExpandButtonClicked
```

Visual Basic Copy Code

```
Public Event ExpandButtonClicked As EventHandler(Of NodeEventArgs)
```

Event Data

ExpandButtonClicked event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode whose [±] button has been clicked.

Remarks

This event is raised only if ExpandButtonAction is set to RaiseEvents. Otherwise either the TreeCollapsed or TreeExpanded event is raised, depending on the state of the node.

Example

The following sample code hides or shows all subordinated nodes in a group when the master node's [±] button is clicked:

C# Copy Code

```
void diagram_ExpandButtonClicked(object sender, NodeEventArgs e)
{
    if (e.Node.SubordinateGroup != null)
    {
        foreach (DiagramItem item in e.Node.SubordinateGroup.AttachedNodes)
            item.Visible = e.Node.Expanded;
    }
}
```

Visual Basic Copy Code

```
Sub diagram_ExpandButtonClicked(ByVal sender As Object, ByVal e As NodeEventArgs) Handles
diagram.ExpandButtonClicked

    If Not e.Node.SubordinateGroup Is Nothing Then

        Dim item As DiagramItem
        For Each item In e.Node.SubordinateGroup.AttachedNodes
            item.Visible = e.Node.Expanded
        Next

    End If

End Sub
```

See Also

DiagramBase Members
DiagramBase Class

MindFusion.Diagramming Namespace

DiagramBase.GroupDestroyed Event

Raised when a Group object is destroyed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<GroupEventArgs> GroupDestroyed
```

Visual Basic Copy Code

```
Public Event GroupDestroyed As EventHandler(Of GroupEventArgs)
```

Event Data

GroupDestroyed event handlers receive an argument of type GroupEventArgs. The following **GroupEventArgs** members provide information relevant to the event:

Member name	Description
Group	The Group that has been destroyed.

Remarks

A group can be destroyed in several ways:

- by removing it from the diagram's Groups collection.
- by deleting the group's MainItem.
- in case the main item is a link, if the control point or segment to which subordinated items are attached is removed. That could happen if link's Shape or SegmentCount properties change.

When a group is destroyed, by default the group items are not deleted automatically. You could delete such items from the **GroupDestroyed** handler, or enable AutoDeleteItems to delete them automatically when the group is destroyed.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.HitTestAdjustmentHandles Event

Raised when custom hit testing of item's adjustment handles must be performed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<HitTestEventArgs> HitTestAdjustmentHandles
```

Visual Basic Copy Code

```
Public Event HitTestAdjustmentHandles As EventHandler(Of HitTestEventArgs)
```

Event Data

HitTestAdjustmentHandles event handlers receive an argument of type HitTestEventArgs. The following **HitTestEventArgs** members provide information relevant to the event:

Member name	Description
Item	A DiagramItem whose adjustment handles should be hit tested.
MousePosition	A **Point** specifying the mouse cursor position.
HitResult	Set this property to an AdjustmentHandle instance representing the handle that has been hit.

Remarks

For links, set the result to a LinkAdjustmentHandle initialized with the index of the link's control point that has been hit. For nodes, set the result to a NodeAdjustmentHandle initialized with a NodeHandleType enumeration member.

Example

C# Copy Code

```
void diagram_HitTestAdjustmentHandles(object sender, HitTestEventArgs e)
{
    ShapeNode shapeNode = e.Item as ShapeNode;
    if (shapeNode != null)
    {
        if (shapeNode.Bounds.Contains(e.MousePosition))
        {
            // Always move
            e.HitResult = new NodeAdjustmentHandle(NodeHandleType.Move);
        }
    }
}
```

Visual Basic Copy Code

```
Private Sub diagram_HitTestAdjustmentHandles(ByVal sender As Object, ByVal e As
HitTestEventArgs) Handles diagram.HitTestAdjustmentHandles

    Dim shapeNode As ShapeNode = CType(e.Item, ShapeNode)

    If Not shapeNode Is Nothing Then

        If (shapeNode.Bounds.Contains(e.MousePosition)) Then

            ' Always move
            e.HitResult = New NodeAdjustmentHandle(NodeHandleType.Move)

        End If

    End If

End Sub
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DrawAdjustmentHandles Event

DiagramBase.InitializeLasso Event

Raised when the user starts drawing a selection lasso, before it is rendered on screen.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<SelectionEventArgs> InitializeLasso
```

Visual Basic Copy Code

```
Public Event InitializeLasso As EventHandler(Of SelectionEventArgs)
```

Remarks

This event is raised immediately after the user starts drawing the lasso tool.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.InitializeLink Event

Raised when the user starts drawing a new link, just after the DiagramLink instance is created.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> InitializeLink
```

Visual Basic Copy Code

```
Public Event InitializeLink As EventHandler(Of LinkEventArgs)
```

Event Data

InitializeLink event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The new DiagramLink instance.

Remarks

This event is raised immediately after the user starts drawing a new link. It lets you set the appearance properties of the link if you need their values to be different from the default ones defined in the Diagram object, or if they don't have correspondent properties in the **Diagram** class. The Initialize* events are intended only for initializing the appearance of items. Do not alter the structure of the diagram from their handlers, that is, do not create new items nor delete existing ones; if you do that, the control will be in an undefined state and will likely throw an exception at some point.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkCreating Event
LinkCreated Event
ActionRedone Event

DiagramBase.InitializeNode Event

Raised when the user starts drawing a new node, just after the DiagramNode instance is created.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> InitializeNode
```

Visual Basic Copy Code

```
Public Event InitializeNode As EventHandler(Of NodeEventArgs)
```

Event Data

InitializeNode event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The new DiagramNode instance.

Remarks

This event is raised immediately after the user starts drawing a new node. It lets you set the appearance properties of the node if you need their values to be different from the default ones defined in the Diagram object, or if they don't have correspondent properties in the **Diagram** class. The Initialize* events are intended only for initializing the appearance of items. Do not alter the structure of the diagram from their handlers, that is, do not create new items nor delete existing ones; if you do that, the control will be in an undefined state and will likely throw an exception at some point.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeCreating Event
NodeCreated Event
ActionRedone Event

DiagramBase.ItemAdded Event

Raised when an item is added to the diagram, either programmatically or interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<ItemEventArgs> ItemAdded
```

Visual Basic Copy Code

```
Public Event ItemAdded As EventHandler(Of ItemEventArgs)
```

Event Data

ItemAdded event handlers receive an argument of type ItemEventArgs. The following **ItemEventArgs** members provide information relevant to the event:

Member name	Description
Item	The DiagramItem that has been added to the diagram.

See Also

DiagramBase Members
DiagramBase Class

DiagramBase.ItemRemoved Event

Raised when an item is removed from the diagram, either programmatically or interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<ItemEventArgs> ItemRemoved
```

Visual Basic Copy Code

```
Public Event ItemRemoved As EventHandler(Of ItemEventArgs)
```

Event Data

ItemRemoved event handlers receive an argument of type ItemEventArgs. The following **ItemEventArgs** members provide information relevant to the event:

Member name	Description
Item	The DiagramItem that has been removed from the diagram.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ItemAdded Event

DiagramBase.JsonDeserializeTag Event

Raised when the Tag of a diagram item must be deserialized from JSON.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<JsonSerializeTagEventArgs> JsonDeserializeTag
```

Visual Basic Copy Code

```
Public Event JsonDeserializeTag As EventHandler(Of JsonSerializeTagEventArgs)
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.JsonSerializeTag Event

Raised when the Tag of a diagram item must be serialized to JSON.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<JsonSerializeTagEventArgs> JsonSerializeTag
```

Visual Basic Copy Code

```
Public Event JsonSerializeTag As EventHandler(Of JsonSerializeTagEventArgs)
```

Remarks

You can handle this event to serialize non-primitive Id or Tag objects. If not handled, the component uses **System.Text.Json.JsonSerializer** to get JSON representation of current object.

Example

The following code shows how to serialize a simple structure as a JsonObject. **JsonObject** derives from **Dictionary<string, JsonValue>**, where JsonValue is a boxing class containing typecasting operators for primitive types and **JsonObject** itself (so you could nest objects).

C# Copy Code

```
public class MyTag
{
    public string Name { get; set; }
    public double Value { get; set; }
    public bool Flag { get; set; }
}

static public class MyTagExtensions
{
    static public JsonObject WriteJson(this MyTag tag)
    {
        return new JsonObject
        {
            { "Name", tag.Name },
            { "Value", tag.Value},
            { "Flag", tag.Flag }
        };
    }

    static public void ReadJson(this MyTag tag, JsonObject json)
    {
        tag.Name = json["Name"];
        tag.Value = json["Value"];
        tag.Flag = json["Flag"];
    }
}

void OnJsonSerializeTag(object sender, JsonSerializeTagEventArgs e)
{
    if (e.PropertyName == "Tag" && e.Tag is MyTag myTag)
    {
        e.Representation = myTag.WriteJson();
        e.Handled = true;
    }
}

void OnJsonDeserializeTag(object sender, JsonSerializeTagEventArgs e)
{
    if (e.PropertyName == "Tag" && e.Object is DiagramItem item)
    {
        var myTag = new MyTag();
        myTag.ReadJson(e.Representation);
        e.Tag = myTag;
        e.Handled = true;
    }
}
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.LayerVisibilityChanged Event

Raised when the visibility of a Layer changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LayerEventArgs> LayerVisibilityChanged
```

Visual Basic Copy Code

```
Public Event LayerVisibilityChanged As EventHandler(Of LayerEventArgs)
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.LinkActivated Event

Raised when a link becomes the ActiveItem.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkActivated
```

Visual Basic Copy Code

```
Public Event LinkActivated As EventHandler(Of LinkEventArgs)
```

Event Data

LinkActivated event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that has become the ActiveItem.

Remarks

When the event is raised, **ActiveItem** already refers to the activated link. If multiple selection is disabled, the active item is the only item selected by the user; otherwise it is usually the one added last to the selection.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

LinkDeactivated Event
LinkDeselected Event

DiagramBase.LinkClicked Event

Raised when the user clicks a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkClicked
```

Visual Basic Copy Code

```
Public Event LinkClicked As EventHandler(Of LinkEventArgs)
```

Event Data

LinkClicked event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that has been clicked.
MouseButton	The mouse button that has been pressed by the user.
MousePosition	A **PointF** specifying the mouse cursor position.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkHovered Event
LinkDoubleClicked Event

DiagramBase.LinkCreateCancelled Event

Raised when the user cancels link creation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkCreateCancelled
```

Visual Basic Copy Code

```
Public Event LinkCreateCancelled As EventHandler(Of LinkEventArgs)
```

Event Data

LinkCreateCancelled event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description

Link	Gets the link for which the event raised.
MousePosition	Gets or sets the current mouse pointer position.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.LinkCreated Event

Raised when the user draws a new link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkCreated
```

Visual Basic Copy Code

```
Public Event LinkCreated As EventHandler(Of LinkEventArgs)
```

Event Data

LinkCreated event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that has been created.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
InitializeLink Event
LinkCreating Event

DiagramBase.LinkCreating Event

A validation event raised while the user is drawing a new link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkValidationEventArgs> LinkCreating
```

Visual Basic Copy Code

```
Public Event LinkCreating As EventHandler(Of LinkValidationEventArgs)
```

Event Data

LinkCreating event handlers receive an argument of type LinkValidationEventArgs. The following **LinkValidationEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that is being created.
MousePosition	A **PointF** specifying the mouse cursor position.
Destination	The DiagramNode that will be set as the link's Destination if this event is validated.
AnchorIndex	Index of the anchor point to which the link will be connected if this event is validated.
TableRow	Index of the table row to which the link will be connected if this event is validated.
Cancel	Set this property to **true** to prevent the user from creating the link at the current location. That displays the Stop mouse cursor and releasing the button will cancel the operation.
CancelDrag	Call this method to immediately stop the link creation.

Remarks

LinkCreating is raised after each mouse movement while drawing a new link. Setting **Cancel** to **true** will change the mouse cursor to **DisallowCursor** (a Stop icon by default). The event is raised once again when the mouse button is released. Setting **Cancel** to **true** in that case will cancel the link creation.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkCreated Event

DiagramBase.LinkDeactivated Event

Raised when a link is no longer the ActiveItem.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkDeactivated
```

Visual Basic Copy Code

```
Public Event LinkDeactivated As EventHandler(Of LinkEventArgs)
```

Event Data

LinkDeactivated event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	A DiagramLink that was the ActiveItem.

Remarks

LinkDeactivated is raised when the **ActiveItem** changes and before that it referred to a link.

See Also

DiagramBase Members
DiagramBase Class

DiagramBase.LinkDeleted Event

Raised when a link is deleted, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkDeleted
```

Visual Basic Copy Code

```
Public Event LinkDeleted As EventHandler(Of LinkEventArgs)
```

Event Data

LinkDeleted event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that has been deleted.

Remarks

The link is removed from the Items and Links collections before the event is raised. From within an event handler you can still check the values of simple properties such as Text or Tag. Before deleting a node, all links that link to it are deleted too. So there might be several **LinkDeleted** events raised before NodeDeleted is raised.

To prevent users from deleting a link by pressing the DEL key, handle the LinkDeleting event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkCreated Event

DiagramBase.LinkDeleting Event

Raised when the user tries to delete a link, this event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkValidationEventArgs> LinkDeleting
```

Visual Basic Copy Code

```
Public Event LinkDeleting As EventHandler(Of LinkValidationEventArgs)
```

Event Data

LinkDeleting event handlers receive an argument of type LinkValidationEventArgs. The following **LinkValidationEventArgs** members provide information relevant to the event:

Member name	Description
Link	A DiagramLink that the user is trying to delete by pressing the DEL key.
Cancel	Set this property to **true** to prevent the user from deleting the link.

Remarks

When users delete a node, its connecting links are deleted too. However, in that case **LinkDeleting** validation events are not raised, but just NodeDeleting.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkDeleted Event

DiagramBase.LinkDeselected Event

Raised when a link is deselected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkDeselected
```

Visual Basic Copy Code

```
Public Event LinkDeselected As EventHandler(Of LinkEventArgs)
```

Event Data

LinkDeselected event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that has been deselected.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkSelected Event

DiagramBase.LinkDoubleClicked Event

Raised when the user double-clicks a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkDoubleClicked
```

Visual Basic Copy Code

```
Public Event LinkDoubleClicked As EventHandler(Of LinkEventArgs)
```

Event Data

LinkDoubleClicked event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that has been double-clicked.
MouseButton	The mouse button that has been pressed by the user.
MousePosition	A **PointF** specifying the mouse cursor position.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkClicked Event

DiagramBase.LinkHovered Event

Raised when the mouse pointer hovers over a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkHovered
```

Visual Basic Copy Code

```
Public Event LinkHovered As EventHandler(Of LinkEventArgs)
```

Event Data

LinkHovered event handlers receive an argument of type LinkEventArgs.

Remarks

This event fires when the mouse pointer rests over a link for **HoverDelay** milliseconds.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkClicked Event
LinkPointerEnter Event
LinkPointerLeave Event

DiagramBase.LinkModified Event

Raised when the user moves any control point of a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkModified
```

Visual Basic Copy Code

```
Public Event LinkModified As EventHandler(Of LinkEventArgs)
```

Event Data

LinkModified event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that has been modified.
MousePosition	A **PointF** specifying the mouse cursor position.
AdjustmentHandle	Index of the adjustment handle that has been moved.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkModifying Event

DiagramBase.LinkModifyCancelled Event

Raised when the user cancels move or resize operation being done on a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkModifyCancelled
```

Visual Basic Copy Code

```
Public Event LinkModifyCancelled As EventHandler(Of LinkEventArgs)
```

Event Data

LinkModifyCancelled event handlers receive an argument of type LinkEventArgs.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.LinkModifying Event

A validation event raised while the user is modifying a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkValidationEventArgs> LinkModifying
```

Visual Basic Copy Code

```
Public Event LinkModifying As EventHandler(Of LinkValidationEventArgs)
```

Event Data

LinkModifying event handlers receive an argument of type LinkValidationEventArgs. The following **LinkValidationEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that is being modified.
MousePosition	A **PointF** specifying the mouse cursor position.
AdjustmentHandle	Index of the adjustment handle that is being moved.
ChangingOrigin	A value indicating whether the operation changes the origin of the link.
Origin	The DiagramNode that will be set as the link's Origin if this event is validated.
ChangingDestination	A value indicating whether the operation changes the destination of the link.
Destination	The DiagramNode that will be set as the link's Destination if this event is validated.
AnchorIndex	Index of the anchor point to which the link will be connected if this event is validated.
TableRow	Index of the table row to which the link will be connected if this event is validated.
Cancel	Set this property to **true** to prevent the user from ending the modification at the current location. That displays the Stop mouse cursor and releasing the button will cancel the operation.
CancelDrag	Call this method to immediately stop the link modification.

Remarks

This event is raised after each mouse movement while modifying a link. Setting **Cancel** to **true** will change the mouse cursor to **DisallowCursor** (a Stop icon by default). The event is raised once again when the mouse button is released. Setting **Cancel** to **true** in that case will cancel the modification and return the link to its original state.

The **AdjustmentHandle** index corresponds to the link's control point that is being dragged.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkModified Event

DiagramBase.LinkPasted Event

Raised when a link has been pasted from the clipboard.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkPasted
```

Visual Basic Copy Code

```
Public Event LinkPasted As EventHandler(Of LinkEventArgs)
```

Event Data

LinkPasted event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that has been pasted from the clipboard.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DiagramView.PasteFromClipboard Method

DiagramBase.LinkRouted Event

Raised when a link is routed by the auto-routing algorithm.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkRouted
```

Visual Basic Copy Code

```
Public Event LinkRouted As EventHandler(Of LinkEventArgs)
```

Event Data

LinkRouted event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that has been routed.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DiagramLink.AutoRoute Property

DiagramBase.LinkSelected Event

Raised when a link is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkSelected
```

Visual Basic Copy Code

```
Public Event LinkSelected As EventHandler(Of LinkEventArgs)
```

Event Data

LinkSelected event handlers receive an argument of type LinkEventArgs. The following **LinkEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that has been selected.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkDeselected Event

DiagramBase.LinkSelecting Event

Raised when the user tries to select a link, this event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkValidationEventArgs> LinkSelecting
```

Visual Basic Copy Code

```
Public Event LinkSelecting As EventHandler(Of LinkValidationEventArgs)
```

Event Data

LinkSelecting event handlers receive an argument of type LinkValidationEventArgs. The following **LinkValidationEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that the user is trying to select.
MousePosition	A **Point** specifying the mouse cursor position.
SelectionRectangle	The current selection rectangle.

Remarks

LinkSelecting is raised in two distinct cases - when the user clicks a link, or while the user is drawing a selection rectangle. When raised because of a mouse click, the **MousePosition** argument is set to a valid point and the **SelectionRectangle** argument equals to **Empty**. While the user is drawing a selection rectangle, the event is raised for each mouse movement, the **SelectionRectangle** contains the selection coordinates and the **MousePosition** parameter equals to **Empty**.

LinkSelecting is raised while enumerating all items in the z-order collection, so you cannot change the link's ZIndex while handling this event. A selection-related event suitable for changing the z-order of items is SelectionChanged.

See Also

DiagramBase Members
DiagramBase Class

DiagramBase.LinkSplit Event

Raised when the user splits a link interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkEventArgs> LinkSplit
```

Visual Basic Copy Code

```
Public Event LinkSplit As EventHandler(Of LinkEventArgs)
```

Event Data

LinkSplit event handlers receive an argument of type LinkEventArgs.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.LinkStartModifying Event

A validation event raised when the user starts modifying a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkValidationEventArgs> LinkStartModifying
```

Visual Basic Copy Code

```
Public Event LinkStartModifying As EventHandler(Of LinkValidationEventArgs)
```

Event Data

LinkStartModifying event handlers receive an argument of type LinkValidationEventArgs. The following **LinkValidationEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that the user is starting to modify.
MousePosition	A **PointF** specifying the mouse cursor position.
AdjustmentHandle	Index of the adjustment handle that is going to be moved.
Cancel	Set this property to **true** to prevent the user from starting the link modification.

See Also

DiagramBase.LinkTextEdited Event

Raised when the user edits the text of a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<EditLinkTextEventArgs> LinkTextEdited
```

Visual Basic Copy Code

```
Public Event LinkTextEdited As EventHandler(Of EditLinkTextEventArgs)
```

Event Data

LinkTextEdited event handlers receive an argument of type EditLinkTextEventArgs.

Remarks

The Label event argument identifies the LinkLabel that has been edited. **Label** is a **null** reference if the user has edited link's Text value.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
LinkTextEditing Event

DiagramBase.LinkTextEditing Event

A validation event that lets you approve or reject changes made to the text of a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkValidationEventArgs> LinkTextEditing
```

Visual Basic Copy Code

```
Public Event LinkTextEditing As EventHandler(Of LinkValidationEventArgs)
```

Event Data

LinkTextEditing event handlers receive an argument of type LinkValidationEventArgs.

Remarks

The Label event argument identifies the LinkLabel that is being edited. **Label** is a **null** reference if the user is editing link's Text value.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeTextEditing Event
CellTextEditing Event
LinkTextEdited Event

DiagramBase.MeasureUnitChanged Event

Raised when the MeasureUnit property value has changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler MeasureUnitChanged
```

Visual Basic Copy Code

```
Public Event MeasureUnitChanged As EventHandler
```

Event Data

MeasureUnitChanged event handlers receive an argument of type **EventArgs**.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
BoundsChanged Event
ActionRedone Event

DiagramBase.NodeActivated Event

Raised when a node becomes the ActiveItem.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeActivated
```

Visual Basic Copy Code

```
Public Event NodeActivated As EventHandler(Of NodeEventArgs)
```

Event Data

NodeActivated event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that has become the ActiveItem.

Remarks

When this event is raised, **ActiveItem** already refers to the activated node. If multiple selection is disabled, the active item is the only item selected by the user; otherwise usually it is the one added last to the selection.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeDeactivated Event
NodeSelected Event

DiagramBase.NodeClicked Event

Raised when the user clicks a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeClicked
```

Visual Basic Copy Code

```
Public Event NodeClicked As EventHandler(Of NodeEventArgs)
```

Event Data

NodeClicked event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that has been clicked.
MouseButton	The mouse button that has been pressed by the user.
MousePosition	A **PointF** specifying the mouse cursor position.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeDoubleClicked Event
CellClicked Event
NodeHovered Event

DiagramBase.NodeCreateCancelled Event

Raised when the user cancels node creation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeCreateCancelled
```

Visual Basic Copy Code

```
Public Event NodeCreateCancelled As EventHandler(Of NodeEventArgs)
```

Event Data

NodeCreateCancelled event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
MousePosition	Gets or sets the current mouse pointer position.
Node	Gets the node for which the event was raised.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.NodeCreated Event

Raised when the user draws a new node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeCreated
```

Visual Basic Copy Code

```
Public Event NodeCreated As EventHandler(Of NodeEventArgs)
```

Event Data

NodeCreated event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that has been created.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
InitializeNode Event
NodeCreating Event
ActionRedone Event

DiagramBase.NodeCreating Event

A validation event raised while the user is drawing a new node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeValidationEventArgs> NodeCreating
```

Visual Basic Copy Code

```
Public Event NodeCreating As EventHandler(Of NodeValidationEventArgs)
```

Event Data

NodeCreating event handlers receive an argument of type NodeValidationEventArgs. The following **NodeValidationEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that is being created.
MousePosition	A **PointF** specifying the mouse cursor position.
Cancel	Set this property to **true** to prevent the user from creating the node at the current location. That displays the Stop mouse cursor and releasing the button will cancel the operation.
CancelDrag	Call this method to immediately stop the node creation.

Remarks

NodeCreating is raised after each mouse movement while drawing a new node. Setting **Cancel** to **true** will change the mouse cursor to **DisallowCursor** (a Stop icon by default). The event is raised once again when the mouse button is released. Setting **Cancel** to **true** in that case will cancel the node creation.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeCreated Event

DiagramBase.NodeDeactivated Event

Raised when a node is no longer the ActiveItem.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeDeactivated
```

Visual Basic Copy Code

```
Public Event NodeDeactivated As EventHandler(Of NodeEventArgs)
```

Event Data

NodeDeactivated event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that was the ActiveItem.

Remarks

NodeDeactivated is raised when the **ActiveItem** changes and before that it referred to a node.

See Also

DiagramBase Members
DiagramBase Class

DiagramBase.NodeDeleted Event

Raised when a node is deleted, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeDeleted
```

Visual Basic Copy Code

```
Public Event NodeDeleted As EventHandler(Of NodeEventArgs)
```

Event Data

NodeDeleted event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that has been deleted.

Remarks

When the event is raised, the node is already removed from the Items and Nodes collections and all links connected to the node are deleted. From within an event handler you can still check the values of simple properties of the node such as Text or Tag.

To prevent users from deleting a node by pressing the DEL key, handle the NodeCreating event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeDeleting Event
NodeCreated Event

DiagramBase.NodeDeleting Event

Raised when the user tries to delete a node, this event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeValidationEventArgs> NodeDeleting
```

Visual Basic Copy Code

```
Public Event NodeDeleting As EventHandler(Of NodeValidationEventArgs)
```

Event Data

NodeDeleting event handlers receive an argument of type NodeValidationEventArgs. The following **NodeValidationEventArgs** members provide information relevant to the event:

Member name	Description
Node	A DiagramNode that the user is trying to delete by pressing the DEL key.
Cancel	Set this property to **true** to prevent the user from deleting the node.

Remarks

When the user deletes a node, the links connected to it are deleted too. However, the LinkDeleting validation event is not raised in that case.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeDeleted Event

DiagramBase.NodeDeselected Event

Raised when a node is deselected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeDeselected
```

Visual Basic Copy Code

```
Public Event NodeDeselected As EventHandler(Of NodeEventArgs)
```

Event Data

NodeDeselected event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that has been deselected.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeSelected Event

DiagramBase.NodeDoubleClicked Event

Raised when the user double-clicks a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeDoubleClicked
```

Visual Basic Copy Code

```
Public Event NodeDoubleClicked As EventHandler(Of NodeEventArgs)
```

Event Data

NodeDoubleClicked event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that has been double-clicked.
MouseButton	The mouse button that has been pressed by the user.
MousePosition	A **Point** specifying the mouse cursor position.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeClicked Event
CellDoubleClicked Event

DiagramBase.NodeHovered Event

Raised when the mouse pointer hovers over a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeHovered
```

Visual Basic Copy Code

```
Public Event NodeHovered As EventHandler(Of NodeEventArgs)
```

Event Data

NodeHovered event handlers receive an argument of type NodeEventArgs.

Remarks

This event fires when the mouse pointer rests over a node for **HoverDelay** milliseconds.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeClicked Event
NodePointerEnter Event
NodePointerLeave Event

DiagramBase.NodeModified Event

Raised when the user moves or resizes a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeModified
```

Visual Basic Copy Code

```
Public Event NodeModified As EventHandler(Of NodeEventArgs)
```

Event Data

NodeModified event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that has been modified.
MousePosition	A **PointF** specifying the mouse cursor position.
AdjustmentHandle	The adjustment handle that has been moved.

Remarks

NodeModified is not raised if the node position is changed programmatically. The event occurs only when the user has dragged an adjustment handle to a new position.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeModifying Event

DiagramBase.NodeModifyCancelled Event

Raised when the user cancels move or resize operation being done on a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeModifyCancelled
```

Visual Basic Copy Code

```
Public Event NodeModifyCancelled As EventHandler(Of NodeEventArgs)
```

Event Data

NodeModifyCancelled event handlers receive an argument of type NodeEventArgs.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.NodeModifying Event

A validation event raised while the user is moving or resizing a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeValidationEventArgs> NodeModifying
```

Visual Basic Copy Code

```
Public Event NodeModifying As EventHandler(Of NodeValidationEventArgs)
```

Event Data

NodeModifying event handlers receive an argument of type NodeValidationEventArgs. The following **NodeValidationEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that is being modified.
MousePosition	A **PointF** specifying the mouse cursor position.
AdjustmentHandle	The adjustment handle that is being moved.
Cancel	Set this property to **true** to prevent the user from ending the modification at the current location. That displays the Stop mouse cursor and releasing the button will cancel the operation.
CancelDrag	Call this method to immediately stop the node modification.

Remarks

This event is raised after each mouse movement while modifying a node. Setting **Cancel** to **true** will change the mouse cursor to **DisallowCursor** (a Stop icon by default). The event is raised once again when the mouse button is released. Setting **Cancel** to **true** in that case will cancel the modification and return the node to its original location.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeModified Event

DiagramBase.NodePasted Event

Raised when a node has been pasted from the clipboard.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodePasted
```

Visual Basic Copy Code

```
Public Event NodePasted As EventHandler(Of NodeEventArgs)
```

Event Data

NodePasted event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that has been pasted from the clipboard.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DiagramView.PasteFromClipboard Method

DiagramBase.NodeSelected Event

Raised when a node is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> NodeSelected
```

Visual Basic Copy Code

```
Public Event NodeSelected As EventHandler(Of NodeEventArgs)
```

Event Data

NodeSelected event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that has been selected.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeDeselected Event

DiagramBase.NodeSelecting Event

Raised when the user tries to select a node, this event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeValidationEventArgs> NodeSelecting
```

Visual Basic Copy Code

```
Public Event NodeSelecting As EventHandler(Of NodeValidationEventArgs)
```

Event Data

NodeSelecting event handlers receive an argument of type NodeValidationEventArgs. The following **NodeValidationEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that the user is trying to select.
MousePosition	A **Point** specifying the mouse cursor position.
SelectionRectangle	The current selection rectangle.

Remarks

NodeSelecting is raised in two distinct cases - when the user clicks a node, or while the user is drawing a selection rectangle. When raised because of a mouse click, the **MousePosition** argument is set to a valid point and the **SelectionRectangle** argument equals to **Empty**. While the user is drawing a selection rectangle, the event is raised for each mouse movement, the **SelectionRectangle** contains the selection coordinates and the **MousePosition** parameter equals to **Empty**.

NodeSelecting is raised while enumerating all items in the z-order collection, so you cannot change the node's ZIndex while handling this event. A selection-related event suitable for changing the z-order of items is SelectionChanged.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeSelected Event

DiagramBase.NodeStartModifying Event

A validation event raised when the user starts modifying a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeValidationEventArgs> NodeStartModifying
```

Visual Basic Copy Code

```
Public Event NodeStartModifying As EventHandler(Of NodeValidationEventArgs)
```

Event Data

NodeStartModifying event handlers receive an argument of type NodeValidationEventArgs. The following **NodeValidationEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that the user is starting to modify.
MousePosition	A **PointF** specifying the mouse cursor position.
AdjustmentHandle	The adjustment handle that is going to be moved.
Cancel	Set this property to **true** to prevent the user from starting the node modification.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeModifying Event
ActionRedone Event

DiagramBase.NodeTextEdited Event

Raised when the user edits the text of a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<EditNodeTextEventArgs> NodeTextEdited
```

Visual Basic Copy Code

```
Public Event NodeTextEdited As EventHandler(Of EditNodeTextEventArgs)
```

Event Data

NodeTextEdited event handlers receive an argument of type EditNodeTextEventArgs. The following **EditNodeTextEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode whose text has been edited.
NewText	The node's new text.
OldText	The node's old text.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeTextEditing Event

DiagramBase.NodeTextEditing Event

A validation event that lets you approve or reject changes made to the text of a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeValidationEventArgs> NodeTextEditing
```

Visual Basic Copy Code

```
Public Event NodeTextEditing As EventHandler(Of NodeValidationEventArgs)
```

Event Data

NodeTextEditing event handlers receive an argument of type NodeValidationEventArgs. The following **NodeValidationEventArgs** members provide information relevant to the event:

Member name	Description
Node	The DiagramNode that the user is trying to edit.
Cancel	Set this property to **true** to prevent the user from editing the node's text.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
NodeTextEdited Event
LinkTextEditing Event

DiagramBase.OriginAnchorChanged Event

Raised when the OriginAnchor property value of a link has changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<AnchorPointEventArgs> OriginAnchorChanged
```

Visual Basic Copy Code

```
Public Event OriginAnchorChanged As EventHandler(Of AnchorPointEventArgs)
```

Event Data

OriginAnchorChanged event handlers receive an argument of type AnchorPointEventArgs. The following **AnchorPointEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink whose OriginAnchor has changed.
OldAnchorIndex	Index of the old anchor point.
NewAnchorIndex	Index of the new anchor point.

Remarks

This event is raised every time the **OriginAnchor** property of a link changes, either because a user connects the link end to a different anchor point, or because an application sets it programmatically.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DestinationAnchorChanged Event

DiagramBase.PropertyChanged Event

INotifyPropertyChanged.PropertyChanged implementation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event PropertyChangedEventHandler PropertyChanged
```

Visual Basic Copy Code

```
Public Event PropertyChanged As PropertyChangedEventHandler
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.QueryAlignTarget Event

Raised to validate a node as target for alignment guides when AutoAlignNodes is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeValidationEventArgs> QueryAlignTarget
```

Visual Basic Copy Code

```
Public Event QueryAlignTarget As EventHandler(Of NodeValidationEventArgs)
```

Event Data

Member name	Description
Cancel	Set it to **true** to prevent a node from being used as alignment target.
Node	The DiagramNode that would be used as alignment target.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.Repaint Event

Raised when the diagram views should be repainted.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<RepaintEventArgs> Repaint
```

Visual Basic Copy Code

```
Public Event Repaint As EventHandler(Of RepaintEventArgs)
```

Event Data

Repaint event handlers receive an argument of type RepaintEventArgs. The following **RepaintEventArgs** members provide information relevant to the event:

Member name	Description
InvalidRect	A **Microsoft.Maui.Graphics.Rect** specifying which part of the diagram should be repainted.

Remarks

If **InvalidRect** equals to **Empty**, the whole diagram should be repainted.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
Invalidate Method

DiagramBase.RequestCacheReset Event

Raised when an object within the diagram requests a reset of the cached bitmap.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler RequestCacheReset
```

Visual Basic Copy Code

```
Public Event RequestCacheReset As EventHandler
```

Event Data

RequestCacheReset event handlers receive an argument of type **EventArgs**.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.RequestClipboard Event

Raised when an object within the diagram requests an operation with the clipboard.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<ClipboardEventArgs> RequestClipboard
```

Visual Basic Copy Code

```
Public Event RequestClipboard As EventHandler(Of ClipboardEventArgs)
```

Event Data

RequestClipboard event handlers receive an argument of type ClipboardEventArgs. The following **ClipboardEventArgs** members provide information relevant to the event:

Member name	Description
IsGetOperation	Indicates whether getting or setting the clipboard.
Source	The object, which initiated the operation.

Data	The clipboard data.
DataType	The type of the clipboard data. This is relevant during Clipboard *get* operations.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.RequestDrawInteractions Event

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<DiagramEventArgs> RequestDrawInteractions
```

Visual Basic Copy Code

```
Public Event RequestDrawInteractions As EventHandler(Of DiagramEventArgs)
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.RequestTimer Event

Raised when an object within the diagram needs to start or stop a timer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<TimerEventArgs> RequestTimer
```

Visual Basic Copy Code

```
Public Event RequestTimer As EventHandler(Of TimerEventArgs)
```

Event Data

RequestTimer event handlers receive an argument of type TimerEventArgs.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.SelectionChanged Event

Raised when an item is added to or removed from the selection

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler SelectionChanged
```

Visual Basic Copy Code

```
Public Event SelectionChanged As EventHandler
```

Event Data

SelectionChanged event handlers receive an argument of type **EventArgs**.

Remarks

Use the Nodes and Links properties of the diagram's Selection to find out which items are currently selected.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.SelectionModified Event

Raised when the user moves multiple selected items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<SelectionEventArgs> SelectionModified
```

Visual Basic Copy Code

```
Public Event SelectionModified As EventHandler(Of SelectionEventArgs)
```

Event Data

SelectionModified event handlers receive an argument of type SelectionEventArgs.

Remarks

Use the Nodes and Links properties of the diagram's Selection to find out which items have been moved.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
SelectionModifying Event

DiagramBase.SelectionModifying Event

Raised while the user is moving multiple selected items, this event lets you cancel the modification.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<SelectionValidationEventArgs> SelectionModifying
```

Visual Basic Copy Code

```
Public Event SelectionModifying As EventHandler(Of SelectionValidationEventArgs)
```

Event Data

SelectionModifying event handlers receive an argument of type SelectionValidationEventArgs. The following **SelectionValidationEventArgs** members provide information relevant to the event:

Member name	Description
Cancel	Set this property to **true** to prevent the user from placing the selected items at the current location. That displays the Stop mouse cursor and releasing the button will cancel the operation.
CancelDrag	Call this method to immediately stop the modification.

Remarks

This event is raised after each mouse movement while modifying a selection of items. Setting Cancel to **true** will change the mouse cursor to **DisallowCursor** (a Stop icon by default). The event is raised once again when the mouse button is released. Setting Cancel to true in that case will cancel the modification and return the items to their original position.

Use the Selection property of the Diagram class to access the selected items.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
SelectionModified Event

DiagramBase.SerializeTag Event

Raised when the Tag of a diagram item must be serialized into an XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<SerializeTagEventArgs> SerializeTag
```

Visual Basic Copy Code

```
Public Event SerializeTag As EventHandler(Of SerializeTagEventArgs)
```

Event Data

SerializeTag event handlers receive an argument of type SerializeTagEventArgs. The following **SerializeTagEventArgs** members provide information relevant to the event:

Member name	Description
Object	The object whose Tag is being serialized. This object can be either a diagram item, group, cell or anchor point.
Tag	The Tag object being serialized.
Representation	An XmlElement instance where the XML representation of the Tag should be saved.
Context	An XmlPersistContext object providing information about the document being serialized, along with methods that help you serialize data to XML.
Handled	Set this flag to **true** if you perform custom tag serialization.

Remarks

Only tags of simple value types can be saved and loaded automatically to/from the XML document. You must provide your own serialization for complex types by implementing this event and DeserializeTag.

Example

C#

Copy Code

```
private void Write(string fileName)
{
    diagram.SerializeTag += new SerializeTagEventHandler(SerializeTag);
    diagram.SaveToXml(fileName);
}

void SerializeTag(object sender, SerializeTagEventArgs e)
{
    e.Handled = true;
    object tag = e.Tag;
    string stringRepresentation = "";

    if (tag != null)
    {
        stringRepresentation = tag.GetType().ToString();
        stringRepresentation += ";";
        stringRepresentation += tag.ToString();
    }

    XmlNode repNode =
        e.Context.XmlDocument.CreateTextNode(stringRepresentation);
    e.Representation.AppendChild(repNode);
}
```

Visual Basic

Copy Code

```
Private Sub Write(ByVal fileName As String)

    AddHandler diagram.SerializeTag, New SerializeTagEventHandler(AddressOf SerializeTag)
    diagram.SaveToXml(fileName)

End Sub

Sub SerializeTag(ByVal sender As Object, ByVal e As SerializeTagEventArgs)

    e.Handled = True

    Dim tag As Object = e.Tag
    Dim stringRepresentation As String = ""

    If Not Tag Is Nothing Then

        stringRepresentation = Tag.GetType().ToString()
        stringRepresentation += ";"
        stringRepresentation += Tag.ToString()

    End If
```

```
    Dim repNode As XmlNode = _
        e.Context.XmlDocument.CreateTextNode(stringRepresentation)
    e.Representation.AppendChild(repNode)

End Sub
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DeserializeTag Event
Diagram.SaveToXml Method

DiagramBase.SetSelfLoopShape Event

Raised when a link becomes a self-loop, giving you a chance to override the default look shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkShapeEventArgs> SetSelfLoopShape
```

Visual Basic Copy Code

```
Public Event SetSelfLoopShape As EventHandler(Of LinkShapeEventArgs)
```

Event Data

SetSelfLoopShape event handlers receive an argument of type LinkShapeEventArgs. The following **LinkShapeEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that becomes a self-loop.
Handled	Indicates whether the event handler has set a custom loop shape.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.ShapeModified Event

Raised when the user modifies the shape of a ShapeNode by moving one of the control points corresponding to shape formula parameters.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<ControlPointEventArgs> ShapeModified
```

Visual Basic Copy Code

```
Public Event ShapeModified As EventHandler(Of ControlPointEventArgs)
```

Event Data

ShapeModified event handlers receive an argument of type ControlPointEventArgs. The following **ControlPointEventArgs** members provide information relevant to the event:

Member name	Description
ControlPoint	Gets the ShapeControlPoint used to modify the node's shape.
NewValue	Gets the new position of the control point.
Node	Gets the ShapeNode whose Shape has been modified.
OldValue	Gets the original position of the control point.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.ShapeNotDefined Event

Raised when the diagram being loaded contains nodes with an undefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<ShapeEventArgs> ShapeNotDefined
```

Visual Basic Copy Code

```
Public Event ShapeNotDefined As EventHandler(Of ShapeEventArgs)
```

Event Data

The UnknownId argument contains the shape identifier. Set the Shape argument to assign to the loaded node.

Remarks

You can handle this event by loading a ShapeLibrary file or creating a new Shape instance from code.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.TableColumnResized Event

Raised when the user finishes resizing a column in a table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<CellEventArgs> TableColumnResized
```

Visual Basic Copy Code

```
Public Event TableColumnResized As EventHandler(Of CellEventArgs)
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.TableColumnResizing Event

Raised when the user resizes a column in a table interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<CellValidationEventArgs> TableColumnResizing
```

Visual Basic Copy Code

```
Public Event TableColumnResizing As EventHandler(Of CellValidationEventArgs)
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.TableRowResized Event

Raised when the user finishes resizing a row in a table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<CellEventArgs> TableRowResized
```

Visual Basic Copy Code

```
Public Event TableRowResized As EventHandler(Of CellEventArgs)
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.TableRowResizing Event

Raised when the user resizes a row in a table interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<CellValidationEventArgs> TableRowResizing
```

Visual Basic Copy Code

```
Public Event TableRowResizing As EventHandler(Of CellValidationEventArgs)
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.TableSectionCollapsed Event

Raised when the user collapses the rows of a table's section by clicking the [-] button in the section's header row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<RowEventArgs> TableSectionCollapsed
```

Visual Basic Copy Code

```
Public Event TableSectionCollapsed As EventHandler(Of RowEventArgs)
```

Event Data

TableSectionCollapsed event handlers receive an argument of type RowEventArgs. The following **RowEventArgs** members provide information relevant to the event:

Member name	Description
Table	The TableNode that contains the collapsed rows.
RowIndex	Index of the header row of the collapsed section.

Remarks

A header row is a row whose Header property is enabled. Such rows display a [±] button which can be used to expand or collapse all rows below a header up to the next header. **TableSectionCollapsed** is raised when a user clicks the [-] button, but not when Expanded is set to **false** programmatically.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
TableSectionExpanded Event

DiagramBase.TableSectionExpanded Event

Raised when the user expands the rows of a table's section by clicking the [+] button in the section's header row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<RowEventArgs> TableSectionExpanded
```

Visual Basic Copy Code

```
Public Event TableSectionExpanded As EventHandler(Of RowEventArgs)
```

Event Data

TableSectionExpanded event handlers receive an argument of type RowEventArgs. The following **RowEventArgs** members provide information relevant to the event:

Member name	Description
Table	The TableNode that contains the expanded rows.
RowIndex	Index of the header row of the expanded section.

Remarks

A header row is a row whose Header property is enabled. Such rows display a [±] button which can be used to expand or collapse all rows below a header up to the next header. **TableSectionExpanded** is raised when a user clicks the [+] button, but not when Expanded is set to **true** programmatically.

Example

The code below changes the color of the first cell of a header row when the table section defined via that row is expanded:

C# Copy Code

```
void diagram_TableSectionExpanded(object sender, RowEventArgs e)
{
    e.Table[0, e.RowIndex].Brush = new MindFusion.Drawing.SolidBrush(Color.Blue);
}
```

Visual Basic Copy Code

```
Private Sub Diagram_TableSectionExpanded(ByVal sender As Object, ByVal e As
MindFusion.Diagramming.RowEventArgs)Handles Diagram.TableSectionExpanded
    e.Table(0, e.RowIndex).Brush = New MindFusion.Drawing.SolidBrush(Color.Blue)
End Sub
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
TableSectionCollapsed Event

DiagramBase.TreeCollapsed Event

Raised when a user collapses a tree branch by clicking the [-] button of an Expandable node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> TreeCollapsed
```

Visual Basic Copy Code

```
Public Event TreeCollapsed As EventHandler(Of NodeEventArgs)
```

Event Data

TreeCollapsed event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The root DiagramNode of the collapsed tree branch.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
TreeExpanded Event
ActionRedone Event

DiagramBase.TreeExpanded Event

Raised when a user expands a tree branch by clicking the [+] button of an Expandable node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NodeEventArgs> TreeExpanded
```

Visual Basic Copy Code

```
Public Event TreeExpanded As EventHandler(Of NodeEventArgs)
```

Event Data

TreeExpanded event handlers receive an argument of type NodeEventArgs. The following **NodeEventArgs** members provide information relevant to the event:

Member name	Description
Node	The root DiagramNode of the expanded tree branch.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
TreeCollapsed Event

DiagramBase.TreeItemTextEditing Event

A validation event that lets you approve or reject changes made to the text of a TreeViewNode's item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<TreeItemValidationEventArgs> TreeItemTextEditing
```

Visual Basic Copy Code

```
Public Event TreeItemTextEditing As EventHandler(Of TreeItemValidationEventArgs)
```

Event Data

TreeItemTextEditing event handlers receive an argument of type TreeItemValidationEventArgs.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.ValidateAnchorPoint Event

A validation event that lets you approve or reject connecting a link to an anchor point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkValidationEventArgs> ValidateAnchorPoint
```

Visual Basic Copy Code

```
Public Event ValidateAnchorPoint As EventHandler(Of LinkValidationEventArgs)
```

Event Data

ValidateAnchorPoint event handlers receive an argument of type LinkValidationEventArgs. The following **LinkValidationEventArgs** members provide information relevant to the event:

Member name	Description
Link	The DiagramLink that will be connected to the anchor point.
MousePosition	A **PointF** specifying the mouse cursor position.
AdjustmentHandle	Index of the adjustment handle that is being moved.
ChangingOrigin	A value indicating whether the operation changes the origin of the link.
ChangingDestination	A value indicating whether the operation changes the destination of the link.
Node	The DiagramNode to which the link will be connected.
AnchorIndex	Index of an anchor point to which the link will be connected if this event is validated.
Cancel	Set this property to **true** to prevent connecting the link to specified point.

Remarks

This event is raised for each point of an AnchorPattern while a link's end point is being moved over a node. If none of the anchor points are validated, the mouse cursor icon changes to **DisallowLinkCursor**, indicating that the link cannot be connected to that node.

When the mouse button is released over a node, the event is raised again for the node's anchor points. From all validated anchor points, the one nearest to the dragged link end point is used to dock the link.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.ValidateLinkCrossing Event

A validation event that lets you approve or reject displaying a decoration at the intersection point of links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<LinkCrossingEventArgs> ValidateLinkCrossing
```

Visual Basic Copy Code

```
Public Event ValidateLinkCrossing As EventHandler(Of LinkCrossingEventArgs)
```

Remarks

This event is raised for each intersection point of links when LinkCrossings property is enabled.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.ViewRemoved Event

Raised when a DiagramView is detached from this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<CancelEventArgs> ViewRemoved
```

Visual Basic Copy Code

```
Public Event ViewRemoved As EventHandler(Of CancelEventArgs)
```

Event Data

ViewRemoved event handlers receive an argument of type **CancelEventArgs**.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase Class

The base class of the Diagram and DiagramDocument classes.

Syntax

JavaScript Copy Code

```
// class
DiagramBase.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.DiagramBase

See Also

DiagramBase Members
MindFusion.Diagramming Namespace

DiagramBase Members

The base class of the Diagram and DiagramDocument classes.

The following tables list the members exposed by the DiagramBase type.

Public Constructors

	Name	Description
	DiagramBase	Initializes a new instance of the DiagramBase class.

Public Methods

	Name	Description
	raiseClicked	Raises an event which notifies that a Diagram item has been clicked.
	raiseCreated	Raises an event which notifies that a Diagram item has been created.
	raiseDeleted	Raises an event which notifies that a Diagram item has been deleted.
	raiseDeselected	Raises an event which notifies that a Diagram item has been deselected.
	raiseDoubleClicked	Raises an event which notifies that a Diagram item has been double-clicked.
	raiseModified	Raises an event which notifies that a Diagram item has been modified.
	raisePasted	Raises an event which notifies that a Diagram item has been pasted from clipboard.
	raiseSelected	Raises an event which notifies that a Diagram item has been selected.
	raiseSelecting	Raises an event which notifies that a Diagram item will be selected.
	raiseSelectingAsync	Raises an event which notifies that a Diagram item will be selected.
	raiseSelectionChanged	Raises an event which notifies that the selection has changed.

Public Events

	Name	Description
	actionRedone	Raised when an action is redone.
	actionUndone	Raised when an action is undone.
	animatedLayoutCompleted	Raised when an animated layout completes.
	cellTextEdited	Raised when the user edits the text of a table cell.
	cellTextEditing	Raised when the user starts to edit the text of a table cell.

	clicked	Raised when the user clicks the diagram at a point where no items are located.
	containerChildAdded	Raised when the user adds a node to a ContainerNode.
	containerChildAdding	A validation event raised to let you prevent users from adding child nodes to a container. Event handlers receive a NodeEventArgs instance that contains data about this event.
	containerChildRemoved	Raised when the user removes a node from a **ContainerNode**.
	containerChildRemoving	Raised when the user removes a node from a **ContainerNode**.
	containerFolded	Raised when the user folds a container by clicking the arrow-up button in the container's caption bar. Event handlers receive a NodeEventArgs instance that contains data about this event.
	containerPageChanged	Raised when the user changes current page of a PagedContainerNode.
	containerUnfolded	Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar. Event handlers receive a NodeEventArgs instance that contains data about this event.
	dataLoaded	Raised when the diagram data is loaded.
	deserializeTag	Raised when the Tag or Id of a diagram item must be deserialized from XML format. Event handlers receive a SerializeTagEventArgs instance that contains data about this event.
	drawAdjustmentHandles	Raised to draw custom adjustment handles.
	drawAnchorPoint	Raised when an anchor point must be custom-drawn.
	expandButtonClicked	Raised if expandButtonAction is set to RaiseEvents and the [+/-]button of an expandable node is clicked.
	headerResized	Raised when the user finished resizing a header.
	headerStartResizing	Raised when the user starts resizing a header.
	hitTestAdjustmentHandles	Raised to hit-test custom adjustment handles.
	hyperlinkClicked	Raised when the user clicks a hyperlink inside a node.
	initializeLink	Raised when the user starts drawing a new link, just after the DiagramLink instance is created.
	initializeNode	Raised when the user starts drawing a new node, just after the DiagramNode instance is created.
	itemAdded	Raised when an item is added to the diagram.
	itemRemoved	Raised when an item is removed from the diagram.
	laneGridCellTextEdited	Raised when the user edits the text of a grid cell.
	linkClicked	Raised when the user clicks a link.
	linkCreated	Raised when the user draws a new link.
	linkCreating	A validation event raised while the user is drawing a new link.
	linkDeleted	Raised when a link is deleted, either programmatically or by the user.
	linkDeleting	Raised when the user tries to delete a link. This event lets you cancel the operation.
	linkDeselected	Raised when a link is deselected, either programmatically or by the user.
	linkDoubleClicked	Raised when the user double-clicks a link.
	linkHovered	Raised when the mouse pointer hovers over a link.
	linkModified	Raised when the user moves any control point of a link.

	linkModifying	A validation event raised while the user is modifying a link.
	linkPasted	Raised when a link has been pasted from the clipboard.
	linkPointerEnter	Raised when the mouse pointer enters the boundaries of a link.
	linkPointerLeave	Raised when the mouse pointer leaves the boundaries of a link.
	linkRouted	Raised when a link is routed by the auto-routing algorithm.
	linkSelected	Raised when a link is selected, either programmatically or by the user.
	linkSelecting	Raised when the user tries to select a link. This event lets you cancel the operation.
	linkSplit	Raised when the user splits a link interactively.
	linkTextEdited	Raised when the user edits the text of a link.
	linkTextEditing	Raised when the user starts to edit the text of a link.
	nodeClicked	Raised when the user clicks a node.
	nodeCreated	Raised when the user draws a new node.
	nodeCreating	A validation event raised while the user is drawing a new node.
	nodeDeleted	Raised when a node is deleted, either programmatically or by the user.
	nodeDeleting	Raised when the user tries to delete a node. This event lets you cancel the operation.
	nodeDeselected	Raised when a node is deselected, either programmatically or by the user.
	nodeDomCreated	Raised when the content of a ControlNode is added to the page DOM.
	nodeDoubleClicked	Raised when the user double-clicks a node.
	nodeHovered	Raised when the mouse pointer hovers over a node.
	nodeModified	Raised when the user moves or resizes a node.
	nodeModifying	A validation event raised while the user is moving or resizing a node.
	nodePasted	Raised when a node has been pasted from the clipboard.
	nodePointerEnter	Raised when the mouse pointer enters the boundaries of a node.
	nodePointerLeave	Raised when the mouse pointer leaves the boundaries of a node.
	nodeSelected	Raised when a node is selected, either programmatically or by the user.
	nodeSelecting	Raised when the user tries to select a node. This event lets you cancel the operation.
	nodeTextEdited	Raised when the user edits the text of a node.
	nodeTextEditing	Raised when the user starts to edit the text of a node.
	queryView	Raised to enumerate diagram views.
	selectionChanged	Raised when the diagram selection has changed.
	selectionModifying	A validation event raised while the user is moving or resizing multiple items.
	serializeTag	Raised when the Tag or Id of a diagram item must be serialized in XML format. Event handlers receive a SerializeTagEventArgs instance that contains data about this event.
	shapeLibraryLoaded	Raised when a ShapeLibrary is loaded from shapeLibraryLocation.
	sizeChanged	Raised when the size of the diagram canvas has changed.

	tableColumnResized	Raised when the user finishes resizing a column in a table.
	tableColumnResizing	Raised when the user resizes a column in a table interactively.
	tableRowResized	Raised when the user finishes resizing a row in a table.
	tableRowResizing	Raised when the user resizes a row in a table interactively.
	tableSectionCollapsed	Raised when a table section is collapsed.
	tableSectionExpanded	Raised when a table section is expanded.
	treeCollapsed	Raised when a user collapses a tree branch by clicking the [-] button of an **expandable** node.
	treeExpanded	Raised when a user expands a tree branch by clicking the [+] button of an **expandable** node.
	treeItemTextEdited	Raised when the user edits the text of a tree view item.
	treeItemTextEditing	Raised when the user starts to edit the text of a tree view item.
	validateAnchorPoint	A validation event that lets you approve or reject connecting a link to an anchor point.
	validateLinkCrossing	A validation event that lets you approve or reject displaying a decoration at the intersection point of links.

See Also

DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase Constructor

Initializes a new instance of the DiagramBase class.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
function DiagramBase ()
```

See Also

DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase Methods

DiagramBase.raiseClicked Method

Raises an event which notifies that a Diagram item has been clicked.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
function raiseClicked (item, mousePosition, mouseButton)
```

Parameters

item
DiagramItem. The Diagram item.

mousePosition
Point. The position of the mouse.

mouseButton
MouseButton. The mouse button, with which the item has been clicked.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.raiseCreated Method

Raises an event which notifies that a Diagram item has been created.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
function raiseCreated (item)
```

Parameters

item
DiagramNode | DiagramLink. The diagram item that has been created.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.raiseDeleted Method

Raises an event which notifies that a Diagram item has been deleted.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
function raiseDeleted (item)
```

Parameters

item
DiagramItem. The DiagramItem that has been deleted.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.raiseDeselected Method

Raises an event which notifies that a Diagram item has been deselected.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
function raiseDeselected (item)
```

Parameters

item
DiagramItem. The Diagram item.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.raiseDoubleClicked Method

Raises an event which notifies that a Diagram item has been double-clicked.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
function raiseDoubleClicked (item, mousePosition, mouseButton)
```

Parameters

item
DiagramItem. The Diagram item.

mousePosition
Point. The position of the mouse.

mouseButton
MouseButton. The mouse button, with which the item has been clicked.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.raiseModified Method

Raises an event which notifies that a Diagram item has been modified.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
function raiseModified (item, mousePosition, adjustmentHandle)
```

Parameters

item
DiagramItem. The DiagramItem that has been modified.

mousePosition
Point. The current mouse pointer position.

adjustmentHandle
Number. Identifies the handle used to modify the item.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.raisePasted Method

Raises an event which notifies that a Diagram item has been pasted from clipboard.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
function raisePasted (item)
```

Parameters

item
DiagramItem. The DiagramItem that has been pasted.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.raiseSelected Method

Raises an event which notifies that a Diagram item has been selected.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
function raiseSelected (item)
```

Parameters

item
DiagramItem. The Diagram item.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.raiseSelecting Method

Raises an event which notifies that a Diagram item will be selected.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

```
JavaScript                                                    Copy Code
function raiseSelecting (item)
```

Parameters

item
DiagramItem. The Diagram item.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.raiseSelectingAsync Method

Raises an event which notifies that a Diagram item will be selected.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

```
JavaScript                                                    Copy Code
function raiseSelectingAsync (item)
```

Parameters

item
DiagramItem. The Diagram item.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.raiseSelectionChanged Method

Raises an event which notifies that the selection has changed.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

```
JavaScript                                                    Copy Code
function raiseSelectionChanged (oldItems, newItems)
```

Parameters

oldItems
Array. The selected items before the change.

newItems
Array. The selected items after the change.

See Also

DiagramBase Members
DiagramBase Class

DiagramBase Events

DiagramBase.actionRedone Event

Raised when an action is redone.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.actionRedone.addEventListener(onActionRedone);

function onActionRedone(sender, args)
{
}
```

Event Data

Event handlers receive an UndoEventArgs instance containing data about this event.

Remarks

This event is raised during calls to redo. It is raised both for top-level commands and for the constituent actions of composite commands.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
actionUndone Event

DiagramBase.actionUndone Event

Raised when an action is undone.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.actionUndone.addEventListener(onActionUndone);

function onActionUndone(sender, args)
{
}
```

Event Data

Event handlers receive an UndoEventArgs instance containing data about this event.

Remarks

This event is raised during calls to undo. It is raised both for top-level commands and for the constituent actions of composite commands.

See Also

DiagramBase.animatedLayoutCompleted Event

Raised when an animated layout completes.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.animatedLayoutCompleted.addEventListener(onAnimatedLayoutCompleted);

function onAnimatedLayoutCompleted(sender, args)
{
}
```

Event Data

When layout is applied on a ContainerNode, event handlers receive a NodeEventArgs instance as argument.

Example

This example handles the **animatedLayoutCompleted** event to resize the diagram.

JavaScript Copy Code

```
diagram.animatedLayoutCompleted.addEventListener(
    (sender, args) =>
    {
        sender.resizeToFitItems(10);
    });
diagram.arrangeAnimated(...);
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
arrangeAnimated Method

DiagramBase.cellTextEdited Event

Raised when the user edits the text of a table cell.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a CellEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowInplaceEdit Property

DiagramBase.cellTextEditing Event

Raised when the user starts to edit the text of a table cell.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a CellEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowInplaceEdit Property

DiagramBase.clicked Event

Raised when the user clicks the diagram at a point where no items are located.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a DiagramEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
linkClicked Event
nodeClicked Event

DiagramBase.containerChildAdded Event

Raised when the user adds a node to a ContainerNode.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.containerChildAdding Event

A validation event raised to let you prevent users from adding child nodes to a container. Event handlers receive a NodeEventArgs instance that contains data about this event.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.containerChildRemoved Event

Raised when the user removes a node from a ContainerNode.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.containerChildRemoving Event

Raised when the user removes a node from a ContainerNode.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.containerFolded Event

Raised when the user folds a container by clicking the arrow-up button in the container's caption bar. Event handlers receive a NodeEventArgs instance that contains data about this event.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ContainerNode.foldable Property

DiagramBase.containerPageChanged Event

Raised when the user changes current page of a PagedContainerNode.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.containerUnfolded Event

Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar. Event handlers receive a NodeEventArgs instance that contains data about this event.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
ContainerNode.foldable Property

DiagramBase.dataLoaded Event

Raised when the diagram data is loaded.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.dataLoaded.addEventListener(onDataLoaded);

function onDataLoaded(sender, args)
{
}
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.deserializeTag Event

Raised when the Tag or Id of a diagram item must be deserialized from XML format. Event handlers receive a SerializeTagEventArgs instance that contains data about this event.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a SerializeTagEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.drawAdjustmentHandles Event

Raised to draw custom adjustment handles.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.drawAdjustmentHandles.addEventListener(onDrawAdjustmentHandles);

function onDrawAdjustmentHandles(sender, args)
{
}
```

Event Data

Event handlers receive a NodeEventArgs or LinkEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DiagramNode.handlesStyle Property

DiagramBase.drawAnchorPoint Event

Raised when an anchor point must be custom-drawn.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.drawAnchorPoint.addEventListener(onDrawAnchorPoint);

function onDrawAnchorPoint(sender, args)
```

```
{
}
```

Event Data

Event handlers receive a DrawAnchorPointEventArgs instance that contains data about this event.

Remarks

Set the markStyle property of AnchorPoint to Custom to get this event raised.

Example

JavaScript Copy Code

```
function onDrawAnchorPoint(sender, e)
{
    var p = e.location;
    var ctx = e.context;
    ctx.fillStyle = "red";
    ctx.fillRect(p.x - 1, p.y - 1, 2, 2);
}
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.expandButtonClicked Event

Raised if expandButtonAction is set to RaiseEvents and the [+/-]button of an expandable node is clicked.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.expandButtonClicked.addEventListener(onExpandButtonClicked);

function onExpandButtonClicked(sender, args)
{
}
```

Event Data

Event handlers receive a NodeEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.headerResized Event

Raised when the user finished resizing a header.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a HeaderResizeEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.headerStartResizing Event

Raised when the user starts resizing a header.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a HeaderResizeEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.hitTestAdjustmentHandles Event

Raised to hit-test custom adjustment handles.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.hitTestAdjustmentHandles.addEventListener(onHitTestAdjustmentHandles);

function onHitTestAdjustmentHandles(sender, args)
{
}
```

Event Data

Event handlers receive a NodeEventArgs or LinkEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
DiagramNode.handlesStyle Property

DiagramBase.hyperlinkClicked Event

Raised when the user clicks a hyperlink inside a node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.hyperlinkClicked.addEventListener(onHyperlinkClicked);

function onHyperlinkClicked(sender, args)
{
}
```

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.initializeLink Event

Raised when the user starts drawing a new link, just after the DiagramLink instance is created.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.initializeNode Event

Raised when the user starts drawing a new node, just after the DiagramNode instance is created.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.initializeNode.addEventListener(onInitializeNode);

function onInitializeNode(sender, args)
{
}
```

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.itemAdded Event

Raised when an item is added to the diagram.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript — Copy Code

```
diagram.itemAdded.addEventListener(onItemAdded);

function onItemAdded(sender, args)
{
}
```

Event Data

Event handlers receive an ItemEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.itemRemoved Event

Raised when an item is removed from the diagram.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript — Copy Code

```
diagram.itemRemoved.addEventListener(onItemRemoved);

function onItemRemoved(sender, args)
{
}
```

Event Data

Event handlers receive an ItemEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.laneGridCellTextEdited Event

Raised when the user edits the text of a grid cell.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a CellEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.linkClicked Event

Raised when the user clicks a link.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
linkHovered Event
linkDoubleClicked Event

DiagramBase.linkCreated Event

Raised when the user draws a new link.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.linkCreating Event

A validation event raised while the user is drawing a new link.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

Example

This example shows how to handle the **linkCreating** event to prevent cycles in the graph.

JavaScript Copy Code

```
// assuming namespaced scripts
var PathFinder = MindFusion.Diagramming.PathFinder;
```

```
diagram.linkCreating.addEventListener(onLinkCreating);

function onLinkCreating(diagram, args)
{
    if (args.destination == null)
    {
        // not pointing to a node yet
        return;
    }

    var pathFinder = new PathFinder(diagram);
    var path = pathFinder.findShortestPath(
        args.destination, args.origin);

    if (path != null)
    {
        // adding this new link would create a cycle
        // [origin]--[dest]--[path internal nodes]--[origin]

        args.cancel = true;
    }
}
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowSelfLoops Property
allowUnconnectedLinks Property

DiagramBase.linkDeleted Event

Raised when a link is deleted, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.linkDeleting Event

Raised when the user tries to delete a link. This event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.linkDeselected Event

Raised when a link is deselected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.linkDoubleClicked Event

Raised when the user double-clicks a link.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.linkHovered Event

Raised when the mouse pointer hovers over a link.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

Remarks

This event fires when the mouse pointer rests over a link for hoverDelay milliseconds.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
linkClicked Event

linkPointerEnter Event
linkPointerLeave Event

DiagramBase.linkModified Event

Raised when the user moves any control point of a link.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.linkModifying Event

A validation event raised while the user is modifying a link.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.linkPasted Event

Raised when a link has been pasted from the clipboard.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.linkPasted.addEventListener(onLinkPasted);

function onLinkPasted(sender, args)
{
}
```

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class

DiagramBase.linkPointerEnter Event

Raised when the mouse pointer enters the boundaries of a link.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
linkHovered Event
linkPointerLeave Event

DiagramBase.linkPointerLeave Event

Raised when the mouse pointer leaves the boundaries of a link.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
linkHovered Event
linkPointerEnter Event

DiagramBase.linkRouted Event

Raised when a link is routed by the auto-routing algorithm.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.linkRouted.addEventListener(onLinkRouted);

function onLinkRouted(sender, args)
{
}
```

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.linkSelected Event

Raised when a link is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.linkSelecting Event

Raised when the user tries to select a link. This event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.linkSplit Event

Raised when the user splits a link interactively.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.linkSplit.addEventListener(onLinkSplit);

function onLinkSplit(sender, args)
{
}
```

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowSplitLinks Property

DiagramBase.linkTextEdited Event

Raised when the user edits the text of a link.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowInplaceEdit Property

DiagramBase.linkTextEditing Event

Raised when the user starts to edit the text of a link.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a LinkEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeClicked Event

Raised when the user clicks a node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

nodeDoubleClicked Event
nodeHovered Event

DiagramBase.nodeCreated Event

Raised when the user draws a new node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

Example

This example handles the **nodeCreated** event to set new node's anchorPattern.

JavaScript Copy Code

```
diagram.nodeCreated.addEventListener(
    (sender, args) =>
    {
        if (sender.defaultShape.id == "Rectangle")
            args.node.anchorPattern = pattern1;
        else
            args.node.anchorPattern = pattern2;
    });
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeCreating Event

A validation event raised while the user is drawing a new node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeDeleted Event

Raised when a node is deleted, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeDeleting Event

Raised when the user tries to delete a node. This event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeDeselected Event

Raised when a node is deselected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeDomCreated Event

Raised when the content of a ControlNode is added to the page DOM.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeDoubleClicked Event

Raised when the user double-clicks a node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeHovered Event

Raised when the mouse pointer hovers over a node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

Remarks

This event fires when the mouse pointer rests over a node for hoverDelay milliseconds.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeModified Event

Raised when the user moves or resizes a node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
nodeClicked Event

nodePointerEnter Event
nodePointerLeave Event

DiagramBase.nodeModifying Event

A validation event raised while the user is moving or resizing a node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodePasted Event

Raised when a node has been pasted from the clipboard.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.nodePasted.addEventListener(onNodePasted);

function onNodePasted(sender, args)
{
}
```

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodePointerEnter Event

Raised when the mouse pointer enters the boundaries of a node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class

DiagramBase.nodePointerLeave Event

Raised when the mouse pointer leaves the boundaries of a node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeSelected Event

Raised when a node is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeSelecting Event

Raised when the user tries to select a node. This event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.nodeTextEdited Event

Raised when the user edits the text of a node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowInplaceEdit Property

DiagramBase.nodeTextEditing Event

Raised when the user starts to edit the text of a node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.queryView Event

Raised to enumerate diagram views.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.queryView.addEventListener(onQueryView);

function onQueryView(sender, args)
{
}
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.selectionChanged Event

Raised when the diagram selection has changed.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a SelectionChangedEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.selectionModifying Event

A validation event raised while the user is moving or resizing multiple items.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a SelectionEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.serializeTag Event

Raised when the Tag or Id of a diagram item must be serialized in XML format. Event handlers receive a SerializeTagEventArgs instance that contains data about this event.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a SerializeTagEventArgs instance that contains data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.shapeLibraryLoaded Event

Raised when a ShapeLibrary is loaded from shapeLibraryLocation.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.sizeChanged Event

Raised when the size of the diagram canvas has changed.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.sizeChanged.addEventListener(onSizeChanged);

function onSizeChanged(sender, args)
{
}
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.tableColumnResized Event

Raised when the user finishes resizing a column in a table.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.tableColumnResized.addEventListener(onTableColumnResized);

function onTableColumnResized(sender, args)
{
}
```

Event Data

Event handlers receive a CellEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowResizeColumns Property

DiagramBase.tableColumnResizing Event

Raised when the user resizes a column in a table interactively.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.tableColumnResizing.addEventListener(onTableColumnResizing);

function onTableColumnResizing(sender, args)
{
}
```

Event Data

Event handlers receive a CellEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowResizeColumns Property

DiagramBase.tableRowResized Event

Raised when the user finishes resizing a row in a table.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.tableRowResized.addEventListener(onTableRowResized);

function onTableRowResized(sender, args)
{
}
```

Event Data

Event handlers receive a CellEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowResizeRows Property

DiagramBase.tableRowResizing Event

Raised when the user resizes a row in a table interactively.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.tableRowResizing.addEventListener(onTableRowResizing);

function onTableRowResizing(sender, args)
{
}
```

Event Data

Event handlers receive a CellEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowResizeRows Property

DiagramBase.tableSectionCollapsed Event

Raised when a table section is collapsed.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.tableSectionCollapsed.addEventListener(onTableSectionCollapsed);

function onTableSectionCollapsed(sender, args)
{
}
```

Event Data

Event handlers receive a CellEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
TableRow.header Property

DiagramBase.tableSectionExpanded Event

Raised when a table section is expanded.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.tableSectionExpanded.addEventListener(onTableSectionExpanded);

function onTableSectionExpanded(sender, args)
{
}
```

Event Data

Event handlers receive a CellEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
TableRow.header Property

DiagramBase.treeCollapsed Event

Raised when a user collapses a tree branch by clicking the [-] button of an expandable node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.treeExpanded Event

Raised when a user expands a tree branch by clicking the [+] button of an expandable node.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a NodeEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.treeItemTextEdited Event

Raised when the user edits the text of a tree view item.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a TreeItemEventArgs instance containing data about this event.

Example

This example handles the **treeItemTextEdited** event to remove items without text.

JavaScript Copy Code

```
diagram.treeItemTextEdited.addEventListener(
    (sender, args) =>
    {
        if (!args.newText)
            args.node.removeItem(args.item);
    });
```

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowInplaceEdit Property

DiagramBase.treeItemTextEditing Event

Raised when the user starts to edit the text of a tree view item.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

Event Data

Event handlers receive a TreeItemEventArgs instance containing data about this event.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace
allowInplaceEdit Property

DiagramBase.validateAnchorPoint Event

A validation event that lets you approve or reject connecting a link to an anchor point.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.validateAnchorPoint.addEventListener(onValidateAnchorPoint);

function onValidateAnchorPoint(sender, args)
{
}
```

Event Data

Event handlers receive a LinkValidationEventArgs instance containing data about this event.

Remarks

This event is raised for each point of an AnchorPattern while a link's end point is being moved over a node. If none of the anchor points are validated, the mouse cursor icon changes to *NotAllowed*, indicating that the link cannot be connected to that node.

When the mouse button is released over a node, the event is raised again for the node's anchor points. From all validated anchor points, the one nearest to the dragged link end point is used to dock the link.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.validateLinkCrossing Event

A validation event that lets you approve or reject displaying a decoration at the intersection point of links.

Namespace: MindFusion.Diagramming
Assembly: DiagramBase.js

Syntax

JavaScript Copy Code

```
diagram.validateLinkCrossing.addEventListener(onValidateLinkCrossing);
```

```
function onValidateLinkCrossing(sender, args)
{
}
```

Event Data

Event handlers receive a LinkCrossingEventArgs instance containing data about this event.

Remarks

This event is raised for each intersection point of links when linkCrossings property is enabled.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramDocument Class

Represents a collection of diagram pages.

Syntax

JavaScript Copy Code

```
// class
DiagramDocument.prototype = {}
```

Remarks

The **DiagramDocument** class represents a collection of diagram pages or sheets, represented by DiagramPage objects. New pages can be added to the document and existing pages can be removed and reordered by using the addPage and removePage methods respectively. The toJson, saveToXml and saveToString methods of **DiagramDocument** save all pages in a single file, and respectively the fromJson, loadFromXml and loadFromString methods load all pages from the file. **DiagramDocument** can also load files created by the serialization methods of Diagram, and will show them in a single page.

DiagramPage is derived from **Diagram** and only adds a few properties, so any code that processes **Diagram** objects will work with **DiagramPage** objects too. A diagram document forwards each event raised by its pages, giving you a chance to use the same handler for events raised by different pages. If a page should implement distinct user interaction or validation logic, it is also possible to attach handlers to the **Diagram** events inherited by **DiagramPage**.

A single **DiagramPage** could be assigned to the diagram property of DiagramView, and that lets you create your own user interface for presenting a list of pages to the user and selecting a current page. The package also includes a TabbedDiagramView that presents all pages of a document in a tabbed user interface where each tab corresponds to a page. The page displayed in the active tab is exposed by the selectedPage property of **TabbedDiagramView**. The active page can be changed interactively by activating its associated tab. New pages can be added and removed to / from the document through buttons in the tab strip and the pages can be rearranged by dragging their associated tabs with the mouse.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramBase
  MindFusion.Diagramming.DiagramDocument

See Also

DiagramDocument Members
MindFusion.Diagramming Namespace

DiagramDocument Members

Represents a collection of diagram pages.

The following tables list the members exposed by the DiagramDocument type.

Public Constructors

	Name	Description

	DiagramDocument	Initializes a new instance of the DiagramDocument class.

Public Properties

	Name	Description
	dirty	Gets or sets a value indicating whether the diagram document has changed since loading it.
	pages	Gets a list of the diagram pages contained in this document.
	XmlOptions	An XmlOptions object providing properties for customizing the XML serialization process. (Inherited from DiagramBase.)

Public Methods

	Name	Description
	addPage	Adds a page to the document.
	fromJson	Deserializes the document from a JSON string.
	fromXmlDocument	Loads document contents from specified XML Document.
	loadFromString	Loads document contents from specified XML or JSON string.
	loadFromXml	Loads the document from an XML file.
	RaiseNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	removePage	Removes the specified page from the document.
	saveToString	Encodes the document contents into an XML or JSON string.
	saveToXml	Saves the document to an XML file.
	toJson	Serializes the document into a JSON string.
	toXmlDocument	Saves the document into an XML Document.
	TrackMouseHover	Specifies whether the diagram view should track mouse hover events. (Inherited from DiagramBase.)

Protected Methods

	Name	Description
	OnActionRecorded	Raises the ActionRecorded event. (Inherited from DiagramBase.)
	OnActionRecording	Raises the ActionRecording event. (Inherited from DiagramBase.)
	OnActionRedone	Raises the ActionRedone event. (Inherited from DiagramBase.)
	OnActionUndone	Raises the ActionUndone event. (Inherited from DiagramBase.)
	OnBeginLoad	Raises the BeginLoad event. (Inherited from DiagramBase.)
	OnBoundsChanged	Raises the BoundsChanged event. (Inherited from DiagramBase.)
	OnCellClicked	Raises the CellClicked event. (Inherited from DiagramBase.)
	OnCellDoubleClicked	Raises the CellDoubleClicked event. (Inherited from DiagramBase.)
	OnCellTextEdited	Raises the CellTextEdited event. (Inherited from DiagramBase.)
	OnCellTextEditing	Raises the CellTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnClicked	Raises the Clicked event. (Inherited from DiagramBase.)

	OnContainerChildAdded	Raises the ContainerChildAdded event. (Inherited from DiagramBase.)
	OnContainerChildRemoved	Raises the ContainerChildRemoved event. (Inherited from DiagramBase.)
	OnContainerFolded	Raises the ContainerFolded event. (Inherited from DiagramBase.)
	OnContainerPageChanged	Raises the ContainerPageChanged event. (Inherited from DiagramBase.)
	OnContainerUnfolded	Raises the ContainerUnfolded event. (Inherited from DiagramBase.)
	OnContentVisibilityChanged	Raises the ContentVisibilityChanged event. (Inherited from DiagramBase.)
	OnDefaultShapeChanged	Raises the DefaultShapeChanged event. (Inherited from DiagramBase.)
	OnDeserializeTag	Raises the DeserializeTag event. (Inherited from DiagramBase.)
	OnDestinationAnchorChanged	Raises the DestinationAnchorChanged event. (Inherited from DiagramBase.)
	OnDirtyChanged	Raises the DirtyChanged event. (Inherited from DiagramBase.)
	OnDoubleClicked	Raises the DoubleClicked event. (Inherited from DiagramBase.)
	OnDrawAdjustmentHandles	Raises the DrawAdjustmentHandles event. (Inherited from DiagramBase.)
	OnDrawAnchorPoint	Raises the DrawAnchorPoint event. (Inherited from DiagramBase.)
	OnDrawBackground	Raises the DrawBackground event. (Inherited from DiagramBase.)
	OnDrawBranchIndicator	Raises the DrawBranchIndicator event. (Inherited from DiagramBase.)
	OnDrawCell	Raises the DrawCell event. (Inherited from DiagramBase.)
	OnDrawForeground	Raises the DrawForeground event. (Inherited from DiagramBase.)
	OnDrawLink	Raises the DrawLink event. (Inherited from DiagramBase.)
	OnDrawNode	Raises the DrawNode event. (Inherited from DiagramBase.)
	OnEndLoad	Raises the EndLoad event. (Inherited from DiagramBase.)
	OnExpandButtonClicked	Raises the ExpandButtonClicked event. (Inherited from DiagramBase.)
	OnGroupDestroyed	Raises the GroupDestroyed event. (Inherited from DiagramBase.)
	OnHitTestAdjustmentHandles	Raises the HitTestAdjustmentHandles event. (Inherited from DiagramBase.)
	OnInitializeLasso	Raises the InitializeLasso event. (Inherited from DiagramBase.)
	OnInitializeLink	Raises the InitializeLink event. (Inherited from DiagramBase.)
	OnInitializeNode	Raises the InitializeNode event. (Inherited from DiagramBase.)
	OnItemAdded	Raises the ItemAdded event. (Inherited from DiagramBase.)
	OnItemRemoved	Raises the ItemRemoved event. (Inherited from DiagramBase.)
	OnJsonDeserializeTag	Raises the JsonDeserializeTag event. (Inherited from DiagramBase.)
	OnJsonSerializeTag	Raises the JsonSerializeTag event. (Inherited from DiagramBase.)
	OnLayerVisibilityChanged	Raises the LayerVisibilityChanged event. (Inherited from DiagramBase.)
	OnLinkActivated	Raises the LinkActivated event. (Inherited from DiagramBase.)
	OnLinkClicked	Raises the LinkClicked event. (Inherited from DiagramBase.)
	OnLinkCreateCancelled	Raises the LinkCreateCancelled event. (Inherited from DiagramBase.)
	OnLinkCreated	Raises the LinkCreated event. (Inherited from DiagramBase.)
	OnLinkCreating	Raises the LinkCreating event. (Inherited from DiagramBase.)

	OnLinkDeactivated	Raises the LinkDeactivated event. (Inherited from DiagramBase.)
	OnLinkDeleted	Raises the LinkDeleted event. (Inherited from DiagramBase.)
	OnLinkDeleting	Raises the LinkDeleting event. (Inherited from DiagramBase.)
	OnLinkDeselected	Raises the LinkDeselected event. (Inherited from DiagramBase.)
	OnLinkDoubleClicked	Raises the LinkDoubleClicked event. (Inherited from DiagramBase.)
	OnLinkHovered	Raises the LinkHovered event. (Inherited from DiagramBase.)
	OnLinkModified	Raises the LinkModified event. (Inherited from DiagramBase.)
	OnLinkModifyCancelled	Raises the LinkModifyCancelled event. (Inherited from DiagramBase.)
	OnLinkModifying	Raises the LinkModifying event. (Inherited from DiagramBase.)
	OnLinkPasted	Raises the LinkPasted event. (Inherited from DiagramBase.)
	OnLinkRouted	Raises the LinkRouted event. (Inherited from DiagramBase.)
	OnLinkSelected	Raises the LinkSelected event. (Inherited from DiagramBase.)
	OnLinkSelecting	Raises the LinkSelecting event. (Inherited from DiagramBase.)
	OnLinkSplit	Raises the LinkSplit event. (Inherited from DiagramBase.)
	OnLinkStartModifying	Raises the LinkStartModifying event. (Inherited from DiagramBase.)
	OnLinkTextEdited	Raises the LinkTextEdited event. (Inherited from DiagramBase.)
	OnLinkTextEditing	Raises the LinkTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnMeasureUnitChanged	Raises the MeasureUnitChanged event. (Inherited from DiagramBase.)
	OnNodeActivated	Raises the NodeActivated event. (Inherited from DiagramBase.)
	OnNodeClicked	Raises the NodeClicked event. (Inherited from DiagramBase.)
	OnNodeCreateCancelled	Raises the NodeCreateCancelled event. (Inherited from DiagramBase.)
	OnNodeCreated	Raises the NodeCreated event. (Inherited from DiagramBase.)
	OnNodeCreating	Raises the NodeCreating event. (Inherited from DiagramBase.)
	OnNodeDeactivated	Raises the NodeDeactivated event. (Inherited from DiagramBase.)
	OnNodeDeleted	Raises the NodeDeleted event. (Inherited from DiagramBase.)
	OnNodeDeleting	Raises the NodeDeleting event. (Inherited from DiagramBase.)
	OnNodeDeselected	Raises the NodeDeselected event. (Inherited from DiagramBase.)
	OnNodeDoubleClicked	Raises the NodeDoubleClicked event. (Inherited from DiagramBase.)
	OnNodeHovered	Raises the NodeHovered event. (Inherited from DiagramBase.)
	OnNodeModified	Raises the NodeModified event. (Inherited from DiagramBase.)
	OnNodeModifyCancelled	Raises the NodeModifyCancelled event. (Inherited from DiagramBase.)
	OnNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	OnNodePasted	Raises the NodePasted event. (Inherited from DiagramBase.)
	OnNodeSelected	Raises the NodeSelected event. (Inherited from DiagramBase.)
	OnNodeSelecting	Raises the NodeSelecting event. (Inherited from DiagramBase.)
	OnNodeStartModifying	Raises the NodeStartModifying event. (Inherited from DiagramBase.)

	OnNodeTextEdited	Raises the NodeTextEdited event. (Inherited from DiagramBase.)
	OnNodeTextEditing	Raises the NodeTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnOriginAnchorChanged	Raises the OriginAnchorChanged event. (Inherited from DiagramBase.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from DiagramBase.)
	OnQueryAlignTarget	Raises the QueryAlignTarget event. (Inherited from DiagramBase.)
	OnRepaint	Raises the Repaint event. (Inherited from DiagramBase.)
	OnRequestClipboard	Raises the RequestClipboard event. (Inherited from DiagramBase.)
	OnRequestDrawInteractions	Raises the RequestDrawInteractions event. (Inherited from DiagramBase.)
	OnRequestTimer	Raises the RequestTimer event. (Inherited from DiagramBase.)
	OnSelectionChanged	Raises the SelectionChanged event. (Inherited from DiagramBase.)
	OnSelectionModified	Raises the SelectionModified event. (Inherited from DiagramBase.)
	OnSelectionModifying	Raises the SelectionModifying event. (Inherited from DiagramBase.)
	OnSerializeTag	Raises the SerializeTag event. (Inherited from DiagramBase.)
	OnSetSelfLoopShape	Raises the SetSelfLoopShape event. (Inherited from DiagramBase.)
	OnShapeModified	Raises the ShapeModified event. (Inherited from DiagramBase.)
	OnShapeNotDefined	Raises the ShapeNotDefined event. (Inherited from DiagramBase.)
	OnTableColumnResized	Raises the TableColumnResized event. (Inherited from DiagramBase.)
	OnTableColumnResizing	Raises the TableColumnResizing event. (Inherited from DiagramBase.)
	OnTableRowResized	Raises the TableRowResized event. (Inherited from DiagramBase.)
	OnTableRowResizing	Raises the TableRowResizing event. (Inherited from DiagramBase.)
	OnTableSectionCollapsed	Raises the TableSectionCollapsed event. (Inherited from DiagramBase.)
	OnTableSectionExpanded	Raises the TableSectionExpanded event. (Inherited from DiagramBase.)
	OnTreeCollapsed	Raises the TreeCollapsed event. (Inherited from DiagramBase.)
	OnTreeExpanded	Raises the TreeExpanded event. (Inherited from DiagramBase.)
	OnTreeItemTextEditing	Raises the TreeItemTextEditing event. (Inherited from DiagramBase.)
	OnValidateAnchorPoint	Raises the ValidateAnchorPoint event. (Inherited from DiagramBase.)
	OnValidateLinkCrossing	Raises the ValidateLinkCrossing event. (Inherited from DiagramBase.)
	OnViewRemoved	Raises the ViewRemoved event. (Inherited from DiagramBase.)

Public Events

	Name	Description
	ActionRecorded	Raised when an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRecording	A validation event raised before an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRedone	Raised when an action is redone. (Inherited from DiagramBase.)
	ActionUndone	Raised when an action is undone. (Inherited from DiagramBase.)

	ActiveLayerChanged	Raised when the diagram's active layer has changed. (Inherited from DiagramBase.)
	BeginLoad	Raised when the component starts loading diagram data from a file or memory stream. (Inherited from DiagramBase.)
	BoundsChanged	Raised when the Bounds property value has changed. (Inherited from DiagramBase.)
	CellClicked	Raised when the user clicks a table's cell. (Inherited from DiagramBase.)
	CellDoubleClicked	Raised when the user double-clicks a table's cell. (Inherited from DiagramBase.)
	CellTextEdited	Raised when the user edits the text of a table's cell. (Inherited from DiagramBase.)
	CellTextEditing	A validation event that lets you approve or reject changes made to the text of a table's cell. (Inherited from DiagramBase.)
	Clicked	Raised when the user clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	ContainerChildAdded	Raised when a node is added to a ContainerNode, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerChildRemoved	Raised when a node is removed from a **ContainerNode**, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerFolded	Raised when the user folds a container by clicking the arrow-up button in the container's caption bar. (Inherited from DiagramBase.)
	ContainerPageChanged	Raised when the user changes current page of a PagedContainerNode. (Inherited from DiagramBase.)
	ContainerUnfolded	Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar. (Inherited from DiagramBase.)
	ContentVisibilityChanged	Raised when the visibility of container's content changes. (Inherited from DiagramBase.)
	DefaultShapeChanged	Raised when the DefaultShape property value has changed. (Inherited from DiagramBase.)
	DeserializeTag	Raised when the Tag of a diagram item must be deserialized from XML format. (Inherited from DiagramBase.)
	DestinationAnchorChanged	Raised when the DestinationAnchor property value of a link has changed. (Inherited from DiagramBase.)
	DirtyChanged	Raised when the **Dirty** property value has changed. (Inherited from DiagramBase.)
	DoubleClicked	Raised when the user double-clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	DrawAdjustmentHandles	Raised when item's adjustment handles must be custom-drawn. (Inherited from DiagramBase.)
	DrawAnchorPoint	Raised when an anchor point must be custom-drawn. (Inherited from DiagramBase.)
	DrawBackground	Raised to let you custom draw the diagram background. (Inherited from DiagramBase.)
	DrawBranchIndicator	Raised when a branch indicator must be custom-drawn. (Inherited from DiagramBase.)
	DrawCell	Raised when a table's cell must be custom drawn. (Inherited from DiagramBase.)
	DrawForeground	Raised to let you custom draw the diagram foreground. (Inherited from DiagramBase.)

	DrawLink	Raised when a diagram link must be custom drawn. (Inherited from DiagramBase.)
	DrawNode	Raised when a diagram node must be custom drawn. (Inherited from DiagramBase.)
	EndLoad	Raised when the component ends loading diagram data from a file or memory stream. (Inherited from DiagramBase.)
	ExpandButtonClicked	Raised when the user clicks the [±] button of an Expandable node. (Inherited from DiagramBase.)
	GroupDestroyed	Raised when a Group object is destroyed. (Inherited from DiagramBase.)
	HitTestAdjustmentHandles	Raised when custom hit testing of item's adjustment handles must be performed. (Inherited from DiagramBase.)
	InitializeLasso	Raised when the user starts drawing a selection lasso, before it is rendered on screen. (Inherited from DiagramBase.)
	InitializeLink	Raised when the user starts drawing a new link, just after the DiagramLink instance is created. (Inherited from DiagramBase.)
	InitializeNode	Raised when the user starts drawing a new node, just after the DiagramNode instance is created. (Inherited from DiagramBase.)
	ItemAdded	Raised when an item is added to the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	ItemRemoved	Raised when an item is removed from the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	JsonDeserializeTag	Raised when the **Tag** of a diagram item must be deserialized from JSON. (Inherited from DiagramBase.)
	JsonSerializeTag	Raised when the **Tag** of a diagram item must be serialized to JSON. (Inherited from DiagramBase.)
	LayerVisibilityChanged	Raised when the visibility of a Layer changes. (Inherited from DiagramBase.)
	LinkActivated	Raised when a link becomes the ActiveItem. (Inherited from DiagramBase.)
	LinkClicked	Raised when the user clicks a link. (Inherited from DiagramBase.)
	LinkCreateCancelled	Raised when the user cancels link creation. (Inherited from DiagramBase.)
	LinkCreated	Raised when the user draws a new link. (Inherited from DiagramBase.)
	LinkCreating	A validation event raised while the user is drawing a new link. (Inherited from DiagramBase.)
	LinkDeactivated	Raised when a link is no longer the **ActiveItem**. (Inherited from DiagramBase.)
	LinkDeleted	Raised when a link is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDeleting	Raised when the user tries to delete a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkDeselected	Raised when a link is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDoubleClicked	Raised when the user double-clicks a link. (Inherited from DiagramBase.)
	LinkHovered	Raised when the mouse pointer hovers over a link. (Inherited from DiagramBase.)
	LinkModified	Raised when the user moves any control point of a link. (Inherited from DiagramBase.)
	LinkModifyCancelled	Raised when the user cancels move or resize operation being done on a link. (Inherited from DiagramBase.)

	LinkModifying	A validation event raised while the user is modifying a link. (Inherited from DiagramBase.)
	LinkPasted	Raised when a link has been pasted from the clipboard. (Inherited from DiagramBase.)
	LinkRouted	Raised when a link is routed by the auto-routing algorithm. (Inherited from DiagramBase.)
	LinkSelected	Raised when a link is selected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkSelecting	Raised when the user tries to select a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkSplit	Raised when the user splits a link interactively. (Inherited from DiagramBase.)
	LinkStartModifying	A validation event raised when the user starts modifying a link. (Inherited from DiagramBase.)
	LinkTextEdited	Raised when the user edits the text of a link. (Inherited from DiagramBase.)
	LinkTextEditing	A validation event that lets you approve or reject changes made to the text of a link. (Inherited from DiagramBase.)
	MeasureUnitChanged	Raised when the MeasureUnit property value has changed. (Inherited from DiagramBase.)
	NodeActivated	Raised when a node becomes the **ActiveItem**. (Inherited from DiagramBase.)
	NodeClicked	Raised when the user clicks a node. (Inherited from DiagramBase.)
	NodeCreateCancelled	Raised when the user cancels node creation. (Inherited from DiagramBase.)
	NodeCreated	Raised when the user draws a new node. (Inherited from DiagramBase.)
	NodeCreating	A validation event raised while the user is drawing a new node. (Inherited from DiagramBase.)
	NodeDeactivated	Raised when a node is no longer the **ActiveItem**. (Inherited from DiagramBase.)
	NodeDeleted	Raised when a node is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDeleting	Raised when the user tries to delete a node, this event lets you cancel the operation. (Inherited from DiagramBase.)
	NodeDeselected	Raised when a node is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDoubleClicked	Raised when the user double-clicks a node. (Inherited from DiagramBase.)
	NodeHovered	Raised when the mouse pointer hovers over a node. (Inherited from DiagramBase.)
	NodeModified	Raised when the user moves or resizes a node. (Inherited from DiagramBase.)
	NodeModifyCancelled	Raised when the user cancels move or resize operation being done on a node. (Inherited from DiagramBase.)
	NodeModifying	A validation event raised while the user is moving or resizing a node. (Inherited from DiagramBase.)
	NodePasted	Raised when a node has been pasted from the clipboard. (Inherited from DiagramBase.)
	NodeSelected	Raised when a node is selected, either programmatically or by the user. (Inherited from DiagramBase.)

	NodeSelecting	Raised when the user tries to select a node, this event lets you cancel the operation. (Inherited from DiagramBase.)
	NodeStartModifying	A validation event raised when the user starts modifying a node. (Inherited from DiagramBase.)
	NodeTextEdited	Raised when the user edits the text of a node. (Inherited from DiagramBase.)
	NodeTextEditing	A validation event that lets you approve or reject changes made to the text of a node. (Inherited from DiagramBase.)
	OriginAnchorChanged	Raised when the OriginAnchor property value of a link has changed. (Inherited from DiagramBase.)
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation. (Inherited from DiagramBase.)
	QueryAlignTarget	Raised to validate a node as target for alignment guides when AutoAlignNodes is enabled. (Inherited from DiagramBase.)
	Repaint	Raised when the diagram views should be repainted. (Inherited from DiagramBase.)
	RequestCacheReset	Raised when an object within the diagram requests a reset of the cached bitmap. (Inherited from DiagramBase.)
	RequestClipboard	Raised when an object within the diagram requests an operation with the clipboard. (Inherited from DiagramBase.)
	RequestDrawInteractions	For internal use. (Inherited from DiagramBase.)
	RequestTimer	Raised when an object within the diagram needs to start or stop a timer. (Inherited from DiagramBase.)
	SelectionChanged	Raised when an item is added to or removed from the selection (Inherited from DiagramBase.)
	SelectionModified	Raised when the user moves multiple selected items. (Inherited from DiagramBase.)
	SelectionModifying	Raised while the user is moving multiple selected items, this event lets you cancel the modification. (Inherited from DiagramBase.)
	SerializeTag	Raised when the **Tag** of a diagram item must be serialized into an XML document. (Inherited from DiagramBase.)
	SetSelfLoopShape	Raised when a link becomes a self-loop, giving you a chance to override the default look shape. (Inherited from DiagramBase.)
	ShapeModified	Raised when the user modifies the shape of a ShapeNode by moving one of the control points corresponding to shape formula parameters. (Inherited from DiagramBase.)
	ShapeNotDefined	Raised when the diagram being loaded contains nodes with an undefined shape. (Inherited from DiagramBase.)
	TableColumnResized	Raised when the user finishes resizing a column in a table. (Inherited from DiagramBase.)
	TableColumnResizing	Raised when the user resizes a column in a table interactively. (Inherited from DiagramBase.)
	TableRowResized	Raised when the user finishes resizing a row in a table. (Inherited from DiagramBase.)
	TableRowResizing	Raised when the user resizes a row in a table interactively. (Inherited from DiagramBase.)
	TableSectionCollapsed	Raised when the user collapses the rows of a table's section by clicking the [-] button in the section's header row. (Inherited from DiagramBase.)
	TableSectionExpanded	Raised when the user expands the rows of a table's section by clicking the [+] button in the section's header row. (Inherited from DiagramBase.)

	TreeCollapsed	Raised when a user collapses a tree branch by clicking the [-] button of an **Expandable** node. (Inherited from DiagramBase.)
	TreeExpanded	Raised when a user expands a tree branch by clicking the [+] button of an **Expandable** node. (Inherited from DiagramBase.)
	TreeItemTextEditing	A validation event that lets you approve or reject changes made to the text of a TreeViewNode's item. (Inherited from DiagramBase.)
	ValidateAnchorPoint	A validation event that lets you approve or reject connecting a link to an anchor point. (Inherited from DiagramBase.)
	ValidateLinkCrossing	A validation event that lets you approve or reject displaying a decoration at the intersection point of links. (Inherited from DiagramBase.)
	ViewRemoved	Raised when a DiagramView is detached from this diagram. (Inherited from DiagramBase.)

See Also

DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument Constructor

Initializes a new instance of the DiagramDocument class.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
function DiagramDocument ()
```

See Also

DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument Methods

DiagramDocument.addPage Method

Adds a page to the document.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
function addPage (page)
```

Parameters

page
DiagramPage. The page to add.

See Also

DiagramDocument Members
DiagramDocument Class

DiagramDocument.fromJson Method

Deserializes the document from a JSON string.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
function fromJson (json)
```

Parameters

json
String. A string created by the toJson method.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.fromXmlDocument Method

Loads document contents from specified XML Document.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
function fromXmlDocument (document)
```

Parameters

document
Document. A DOM Document object containing XML-serialized DiagramDocument contents.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.loadFromString Method

Loads document contents from specified XML or JSON string.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
function loadFromString (str)
```

Parameters

str

String. A string containing the serialized document contents.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.loadFromXml Method

Loads the document from an XML file.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
function loadFromXml (fileUrl, [onLoad, [onError]])
```

Parameters

fileUrl
String. The URL of an XML file where the data should be read from.

onLoad
Optional.
function. A callback that should be invoked if the file is loaded successfully.

onError
Optional.
function. A callback that should be invoked if the file could not be downloaded.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.removePage Method

Removes the specified page from the document.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
function removePage (page)
```

Parameters

page
DiagramPage. The page to be removed.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.saveToString Method

Encodes the document contents into an XML or JSON string.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
function saveToString (format)
```

Parameters

format
SaveToStringFormat. A member of the SaveToStringFormat enumeration.

Return Value

String. A string containing the document's serialized representation.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.saveToXml Method

Saves the document to an XML file.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
function saveToXml (url)
```

Parameters

url
String. A URL specifying where the document's XML should be posted to.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.toJson Method

Serializes the document into a JSON string.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

Return Value

String. A string containing the document's JSON representation.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.toXmlDocument Method

Saves the document into an XML Document.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
function toXmlDocument ()
```

Return Value

Document. A DOM Document object containing XML-serialized DiagramDocument contents.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument Properties

DiagramDocument.dirty Property

Gets or sets a value indicating whether the diagram document has changed since loading it.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
get dirty() {}
set dirty(value) {}
```

Property Value

Boolean

true if the document has changed, or false otherwise.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.pages Property

Gets a list of the diagram pages contained in this document.

Namespace: MindFusion.Diagramming
Assembly: DiagramDocument.js

Syntax

JavaScript Copy Code

```
get pages() {}
set pages(value) {}
```

Property Value

Array

Array of DiagramPage objects.

Remarks

Use the addPage and removePage methods to respectively add or remove pages.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument Events

DiagramEventArgs Class

Contains the arguments passed to handlers of global diagram events such as clicked.

Syntax

JavaScript Copy Code

```
// class
DiagramEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.EventArgs
  MindFusion.Diagramming.DiagramEventArgs

See Also

DiagramEventArgs Members
MindFusion.Diagramming Namespace
Events Class

DiagramEventArgs Members

Contains the arguments passed to handlers of global diagram events such as clicked.

The following tables list the members exposed by the DiagramEventArgs type.

Public Constructors

	Name	Description
	DiagramEventArgs	Initializes a new instance of the DiagramEventArgs class.

Public Properties

	Name	Description
	deltaX	Gets the horizontal scroll amount of a wheel event.
	deltaY	Gets the vertical scroll amount of a wheel event.

	handled	Gets or sets a value indicating whether the event has been handled. (Inherited from EventArgs.)
	mouseButton	Gets which mouse button has been pressed.
	mousePosition	Gets the current mouse pointer position.

See Also

DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs Constructor

Initializes a new instance of the DiagramEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function DiagramEventArgs ()
```

See Also

DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs Properties

DiagramEventArgs.deltaX Property

Gets the horizontal scroll amount of a wheel event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get deltaX() {}
set deltaX(value) {}
```

Property Value

Number

The horizontal scroll amount.

See Also

DiagramEventArgs Members
DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs.deltaY Property

Gets the vertical scroll amount of a wheel event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get deltaY() {}
set deltaY(value) {}
```

Property Value

Number

The vertical scroll amount.

See Also

DiagramEventArgs Members
DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs.mouseButton Property

Gets which mouse button has been pressed.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get mouseButton() {}
```

Property Value

MouseButton

A member of the MouseButton enumeration.

See Also

DiagramEventArgs Members
DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs.mousePosition Property

Gets the current mouse pointer position.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get mousePosition() {}
```

Property Value

Point

A Point instance containing mouse pointer coordinates.

See Also

DiagramItem Class

Represents an item in the diagram model. All classes representing diagram elements derive from **DiagramItem**.

Syntax

JavaScript Copy Code

```
// class
DiagramItem.prototype = {}
```

Remarks

All items in the diagram document - links and nodes - share common base class called **DiagramItem**. Its main function is to provide interface for the most common methods and properties of every diagram element.

The main textual content of an item is set using the text property. Supplementary labels can be created by calling addLabel.

Appearance can be customized using a shared Style object assigned to the style property. Attributes of the shared style can be overridden by setting item's properties such as brush, stroke, font.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem

See Also

DiagramItem Members
MindFusion.Diagramming Namespace

DiagramItem Members

Represents an item in the diagram model. All classes representing diagram elements derive from DiagramItem.

The following tables list the members exposed by the **DiagramItem** type.

Public Constructors

	Name	Description
	DiagramItem	Initializes a new instance of the **DiagramItem** class.

Public Properties

	Name	Description
	brush	Gets or sets an object specifying how to fill the interior of this item.
	container	Gets the parent ContainerNode containing this node.
	cssClass	Gets or sets the CSS class associated with this item.
	cssElementName	Gets the name of the CSS element associated with this item.
	effectiveBrush	Gets the effective brush that should be used to paint this item.
	effectiveCaptionBackBrush	Gets the effective brush that should be used to paint node's caption bar.
	effectiveFont	Gets the effective font that should be used to render this item's text.
	effectiveFontName	Gets the name of the effective font that should be used to render this item's text.

	effectiveFontSize	Gets the size of the effective font that should be used to render this item's text.
	effectiveFontStyle	Gets the style of the effective font that should be used to render this item's text.
	effectiveOpacity	Gets the effective opacity that should be used to paint this item.
	effectiveShadowColor	Gets the effective color that should be used to render this item's shadow.
	effectiveStroke	Gets the effective color that should be used to stroke this item.
	effectiveStrokeDashArray	Gets the effective line dash array that should be applied when stroking this item.
	effectiveStrokeDashOffset	Gets the effective line dash offset that should be applied when stroking this item.
	effectiveStrokeDashStyle	Gets the effective line dash pattern that should be applied when stroking this item.
	effectiveStrokeThickness	Gets the effective line thickness that should be applied when stroking this item.
	effectiveTextColor	Gets the effective text color that should be used to render this item's text.
	effectiveTextStroke	Gets the effective text color that should be used to render this item's text outline.
	effectiveTextStrokeThickness	Gets the effective width of the text outline of this item.
	font	Gets or sets the font used to render item's text.
	graphicsContent	Gets an array of visual elements used to render this item on the canvas.
	hyperLink	Gets or sets the hyperlink associated with this diagram item.
	id	Gets or sets a custom identifier associated with this item.
	ignoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods.
	layoutTraits	Gets an object containing properties specific to individual layout algorithms.
	lineAlignment	Gets or sets the vertical alignment of item's text.
	locked	Gets or sets whether users are allowed to interact with this item.
	opacity	Gets or sets opacity of this item.
	parent	Gets the Diagram to which this item belongs.
	pen	Gets or sets the color used to stroke item's borders.
	selected	Gets or sets a value indicating whether a diagram item is selected.
	shadowColor	Gets or sets the color used to draw item's shadow.
	shadowOffsetX	Gets or sets the horizontal offset of item's shadow.
	shadowOffsetY	Gets or sets the vertical offset of item's shadow.
	stroke	Gets or sets the color used to stroke item's frame.
	strokeDashArray	Gets or sets an array of values specifying the dash pattern.
	strokeDashOffset	Gets or sets the the distance between the line start and the beginning of the dash pattern.
	strokeDashStyle	Gets or sets the line dash pattern applied when stroking item's frame.
	strokeThickness	Gets or sets the line width applied when stroking item's frame.

	style	Gets or sets the style associated with this item.
	tag	Gets or sets custom data tag associated with this item.
	text	Gets or sets the text label displayed by this item.
	textAlignment	Gets or sets the horizontal alignment of item's text.
	textColor	Gets or sets the color used to fill item's text.
	textComponent	Gets the Text object instance used to render item's text.
	textPadding	Gets or sets the padding distance between item's boundaries and text label.
	textStroke	Gets or sets the color used to stroke item's text.
	textStrokeThickness	Gets or sets the outline width of item's text.
	tooltip	Gets or sets tooltip text that should be displayed when the mouse hovers over this item.
	topLevel	Gets if this is a top-level item in the diagram.
	visible	Gets or sets a value indicating whether this item is visible.
	weight	Gets or sets a weight value used by layout and path-finding algorithms.
	zIndex	Gets the z-order position of this item.

Public Methods

	Name	Description
	addCanvasElements	Derived classes must implement this method by adding their graphics primitives to the canvas.elements collection.
	addLabel	Adds a new label to this item.
	allowDrag	Invoked to validate the drag operation.
	animate	Animates the item.
	clone	Creates a copy of this item.
	containsPoint	Determines whether this item contains the specified point.
	endDrag	Invoked when the user stops dragging this item.
	fromJson	Deserializes this item from a JSON object.
	getRepaintBounds	Gets the repaint region for this item, taking into account factors such as pen width, selection handles and shadow.
	hitTest	Returns the diagram item that contains the specified point.
	invalidate	Invalidates the region of the canvas, that matches the item's repaint bounds.
	loadFromXml	Loads the item content from an XML element.
	onRemove	Called when an item is removed from a **Diagram**, either by the user, programmatically or via undo / redo.
	removeCanvasElements	Derived classes must implement this method by removing their graphics primitives from the canvas.elements collection.
	removeLabel	Removes the specified label from this item.
	restoreState	Loads the properties of this diagram item.
	saveLocationState	Saves the state of this **DiagramItem**.

	saveState	Saves the properties of this diagram item.
	saveToXml	Saves the item content into an XML element.
	startDrag	Invoked when the user starts dragging this item.
	toJson	Serializes this item into a JSON object.
	updateCanvasElements	Derived classes must implement this method to update the **item**'s visual elements.
	updateDrag	Invoked while the user drags this item.

See Also

DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem Constructor

Initializes a new instance of the DiagramItem class.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function DiagramItem ([parent])
```

Parameters

parent
Optional.
Diagram. The Diagram from which to obtain default values for the item properties.

See Also

DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem Methods

DiagramItem.addCanvasElements Method

Derived classes must implement this method by adding their graphics primitives to the canvas.elements collection.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function addCanvasElements ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.addLabel Method

Adds a new label to this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript — Copy Code

```
function addLabel (text)
```

Parameters

text
String. The label's text.

Return Value

A LinkLabel or NodeLabel instance, depending on item's type.

Remarks

An item's main text content is specified via its text property. Call **addLabel** if you need to display additional text labels, such as identifiers or hints to the user.

Position of labels can be specified by calling methods such as setCornerPosition for nodes, or setControlPointPosition for links. You can let users move labels interactively by setting the behavior property of DiagramView to MoveLabels.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
ItemLabel Class
removeLabel Method

DiagramItem.allowDrag Method

Invoked to validate the drag operation.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript — Copy Code

```
function allowDrag (ist)
```

Parameters

ist
An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.animate Method

Animates the item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function animate (animation)
```

Parameters

animation
ItemAnimation. An **ItemAnimation** -derived object.

Remarks

Item's animated values are considered transient and do not change actual property values. If you want to assign last value from animation to actual property such as bounds or opacity, set the *keepLastValue* flag in animation parameters as in example below.

Example

JavaScript Copy Code

```
node.animate(
    new PathAnimation(
        [
            new Point(30, 30),
            new Point(50, 30),
            new Point(75, 66),
            new Point(125, 88)
        ],
        {
            keepLastValue: true
        }));
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Animations

DiagramItem.clone Method

Creates a copy of this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function clone ()
```

Return Value

Object. The copied item.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.containsPoint Method

Determines whether this item contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function containsPoint (point)
```

Parameters

point
The Point to check.

Return Value

Boolean. **true** if this item contains the specified point, or **false** otherwise.

Remarks

This method is used during hit-testing to see if the mouse is inside the item.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.endDrag Method

Invoked when the user stops dragging this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function endDrag (ist)
```

Parameters

ist
InteractionState. An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.fromJson Method

Deserializes this item from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function fromJson (json)
```

Parameters

json
Object. An object, containing the data for this item.

See Also

DiagramItem.getRepaintBounds Method

Gets the repaint region for this item, taking into account factors such as pen width, selection handles and shadow.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function getRepaintBounds ()
```

Return Value

A Rect specifying the region that should be repainted when this item is modified.

Remarks

This method is invoked when an item is modified by the user, or when an appearance property of the item changes. The returned region is invalidated and redrawn at the next paint operation.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
bounds Property

DiagramItem.hitTest Method

Returns the diagram item that contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function hitTest (point, [threshold, [onlyNodes, [onlyInteractive]]])
```

Parameters

point
Point. A Point value specifying the logical coordinates of a diagram point.

threshold
Optional.
Number. Specifies the size of the area around the item outlines where the point is still considered inside the item.

onlyNodes
Optional.
Boolean. true to exclude child links from the search.

onlyInteractive
Optional.
Boolean. true to exclude locked items from the search.

Return Value

DiagramItem. The item that contains the specified point.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.invalidate Method

Invalidates the region of the canvas, that matches the item's repaint bounds.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function invalidate ([dirty, [force]])
```

Parameters

dirty
Optional.
Boolean. Optional.

force
Optional.
Boolean. Optional.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.loadFromXml Method

Loads the item content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that contains the item's serialized content.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
saveToXml Method

DiagramItem.onRemove Method

Called when an item is removed from a Diagram, either by the user, programmatically or via undo / redo.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function onRemove ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.removeCanvasElements Method

Derived classes must implement this method by removing their graphics primitives from the canvas.elements collection.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function removeCanvasElements ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.removeLabel Method

Removes the specified label from this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function removeLabel (label)
```

Parameters

label
An ItemLabel instance.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
addLabel Method

DiagramItem.restoreState Method

Loads the properties of this diagram item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function restoreState (state)
```

Parameters

state
Object. An object, containing the properties of this DiagramItem.

Remarks

This method is used by the undo/redo subsystem.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.saveLocationState Method

Saves the state of this DiagramItem.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function saveLocationState ()
```

Return Value

A DiagramItemState instance.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.saveState Method

Saves the properties of this diagram item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function saveState ()
```

Return Value

Object. An object, containing the properties of this DiagramItem.

Remarks

This method is used by the undo/redo subsystem.

See Also

DiagramItem Members
DiagramItem Class

DiagramItem.saveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
loadFromXml Method

DiagramItem.startDrag Method

Invoked when the user starts dragging this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function startDrag (ist)
```

Parameters

ist
An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.toJson Method

Serializes this item into a JSON object.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

Return Value

Object. An object, containing the data for this item.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.updateCanvasElements Method

Derived classes must implement this method to update the item's visual elements.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function updateCanvasElements ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.updateDrag Method

Invoked while the user drags this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
function updateDrag (ist)
```

Parameters

ist

An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem Properties

DiagramItem.brush Property

Gets or sets an object specifying how to fill the interior of this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get brush() {}
set brush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

Remarks

This property can be set to a color string to specify solid fill:

JavaScript Copy Code

```
node.brush = "white";
```

It also accepts objects that specify the type of brush and additional information:

JavaScript Copy Code

```
node1.brush =
    { type: 'SolidBrush', color: '#e0e9e9' };
node2.brush =
    { type: 'LinearGradientBrush', color1: '#e0e9e9', color2: '#616a7f', angle: 90 };
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
stroke Property
opacity Property

DiagramItem.container Property

Gets the parent ContainerNode containing this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get container() {}
```

Property Value

ContainerNode

The parent ContainerNode of this node.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.cssClass Property

Gets or sets the CSS class associated with this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get cssClass() {}
set cssClass(value) {}
```

Property Value

String

A string containing the CSS class.

Remarks

You could set this value to associate CSS rules with a specific item using class selector.

Example

JavaScript Copy Code

```
// JavaScript
node.text = "Attention";
node.cssClass = "urgent";

// CSS
mfd-shapenode.urgent {
    --brush: #ffcccc;
    --stroke: red;
}
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
style Property

DiagramItem.cssElementName Property

Gets the name of the CSS element associated with this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get cssElementName() {}
set cssElementName(value) {}
```

Property Value

String

A string containing the CSS element name for item's type.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveBrush Property

Gets the effective brush that should be used to paint this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveBrush() {}
```

Property Value

Object

A color string or an object literal specifying gradient.

Remarks

This property returns the brush set locally for an item via its brush property, or looks up in the styling hierarchy for a brush set in the item's style, the diagram's style or the current theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
brush Property
style Property

DiagramItem.effectiveCaptionBackBrush Property

Gets the effective brush that should be used to paint node's caption bar.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveCaptionBackBrush() {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveFont Property

Gets the effective font that should be used to render this item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveFont() {}
```

Property Value

Font

An instance of the Font class.

Remarks

This property returns a font set locally for an item via its font property, or constructs a font from the results of effectiveFontName and effectiveFontSize properties.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
font Property
style Property

DiagramItem.effectiveFontName Property

Gets the name of the effective font that should be used to render this item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveFontName() {}
```

Property Value

String

A string specifying the font name.

Remarks

This property looks up in the styling hierarchy for font names set in the item's style, the diagram's style or the current theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveFontSize Property

Gets the size of the effective font that should be used to render this item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveFontSize() {}
```

Property Value

Number

A number specifying the font size.

Remarks

This property looks up in the styling hierarchy for font sizes set in the item's style, the diagram's style or the current theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveFontStyle Property

Gets the style of the effective font that should be used to render this item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveFontStyle() {}
```

Property Value

FontStyle

A combination of FontStyle enumeration members.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveOpacity Property

Gets the effective opacity that should be used to paint this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveOpacity() {}
set effectiveOpacity(value) {}
```

Property Value

Number

A value between 0 and 1 specifying the item's opacity.

Remarks

Returns opacity value or current OpacityAnimation progress.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveShadowColor Property

Gets the effective color that should be used to render this item's shadow.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveShadowColor() {}
```

Property Value

String

A string specifying the item's shadow color.

Remarks

This property returns the shadow color set locally for an item via shadowColor, or looks up in the styling hierarchy for a shadow color set in the item's style, the diagram's style or the current theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveStroke Property

Gets the effective color that should be used to stroke this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveStroke() {}
```

Property Value

String

A string specifying the item's stroke color.

Remarks

This property returns the stroke color set locally for an item via stroke, or looks up in the styling hierarchy for a stroke color set in the item's style, the diagram's style or the current theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveStrokeDashArray Property

Gets the effective line dash array that should be applied when stroking this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveStrokeDashArray() {}
set effectiveStrokeDashArray(value) {}
```

Property Value

Array

An array of numbers specifying the item's stroke dash array.

Remarks

This property returns the array set locally for an item via strokeDashArray, or looks up in the styling hierarchy for a stroke color set in the item's style, the diagram's style or the current theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveStrokeDashOffset Property

Gets the effective line dash offset that should be applied when stroking this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveStrokeDashOffset() {}
set effectiveStrokeDashOffset(value) {}
```

Property Value

Number

A number specifying the item's stroke dash offset.

Remarks

This property returns the offset set locally for an item via strokeDashOffset, or looks up in the styling hierarchy for a stroke color set in the item's style, the diagram's style or the current theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveStrokeDashStyle Property

Gets the effective line dash pattern that should be applied when stroking this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveStrokeDashStyle() {}
```

Property Value

DashStyle

A member of the DashStyle enumeration.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveStrokeThickness Property

Gets the effective line thickness that should be applied when stroking this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveStrokeThickness() {}
```

Property Value

Number

A number specifying the item's stroke thickness.

Remarks

This method returns the stroke thickness set locally for an item via strokeThickness, or looks up in the styling hierarchy for a thickness set in the item's style, the diagram's style or the current theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
strokeThickness Property
style Property

DiagramItem.effectiveTextColor Property

Gets the effective text color that should be used to render this item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveTextColor() {}
```

Property Value

String

A string specifying the item's text color.

Remarks

This method returns the text color set locally for an item via textColor, or looks up in the styling hierarchy for a text color set in the item's style, the diagram's style or the current theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveTextStroke Property

Gets the effective text color that should be used to render this item's text outline.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveTextStroke() {}
```

Property Value

String

A string specifying the item's text outline color.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.effectiveTextStrokeThickness Property

Gets the effective width of the text outline of this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get effectiveTextStrokeThickness() {}
```

Property Value

Number

The width of the text outline of this item.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.font Property

Gets or sets the font used to render item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get font() {}
set font(value) {}
```

Property Value

Font

An instance of the Font class.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

text Property
textColor Property

DiagramItem.graphicsContent Property

Gets an array of visual elements used to render this item on the canvas.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get graphicsContent() {}
```

Property Value

Array

An array containing objects from MindFusion.Drawing namespace.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.hyperLink Property

Gets or sets the hyperlink associated with this diagram item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get hyperLink() {}
set hyperLink(value) {}
```

Property Value

String

A string containing hyperlink URL.

Remarks

This is just a data property, and clicking an item whose **hyperlink** is set does not automatically open the target page. If that's required, you could call **window.open** fron nodeClicked or linkClicked event handlers.

Another way to associate hyperlinks with items is by embedding <a> elements in text when enableStyledText is set to **true**. The control raises hyperlinkClicked event when users click on such embeded hyperlinks.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.id Property

Gets or sets a custom identifier associated with this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get id() {}
set id(value) {}
```

Property Value

Object

An Object containing custom identifier.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.ignoreLayout Property

Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get ignoreLayout() {}
set ignoreLayout(value) {}
```

Property Value

Boolean

true to ignore this item when arranging the diagram, or **false** otherwise.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.layoutTraits Property

Gets an object containing properties specific to individual layout algorithms.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get layoutTraits() {}
```

Property Value

Object

An object containing layout properties.

Remarks

At this time this property is used only by TreeLayout in JavaScript version of the library:

- **TreeLayout** looks for a **treeLayoutAssistant** value set to a member of the AssistantType enumeration.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.lineAlignment Property

Gets or sets the vertical alignment of item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get lineAlignment() {}
set lineAlignment(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.locked Property

Gets or sets whether users are allowed to interact with this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get locked() {}
set locked(value) {}
```

Property Value

Boolean

true to prevent interaction, or **false** otherwise. Default value is **false**.

Remarks

A locked item cannot be selected, moved or resized interactively. Additionally, its text cannot be edited in-place.

Locked nodes still can be moved or resized programmatically through their bounds property. Positions of locked links' control points can be changed programmatically too - through their controlPoints collections.

Locked items positions can be changed also by the automatic layout algorithms. To prevent automatic layout methods from changing positions of individual items, set the ignoreLayout property of DiagramItem to **true**.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
visible Property

DiagramItem.opacity Property

Gets or sets opacity of this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript — Copy Code

```
get opacity() {}
set opacity(value) {}
```

Property Value

Number

A value between 0 and 1 specifying the item's opacity.

Remarks

This value is applied as a factor multiplying the alpha channel of all colors in the item's graphics.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
brush Property
stroke Property

DiagramItem.parent Property

Gets the Diagram to which this item belongs.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript — Copy Code

```
get parent() {}
```

Property Value

Diagram

An instance of the Diagram class.

Remarks

Call diagram's addItem method to add items to it.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.pen Property

Gets or sets the color used to stroke item's borders.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get pen() {}
set pen(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

Remarks

This is synonim for the stroke property, and will be removed in future versions of the library.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.selected Property

Gets or sets a value indicating whether a diagram item is selected.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get selected() {}
set selected(value) {}
```

Property Value

Boolean

true if the item is selected, or **false** otherwise.

Remarks

Use the property to check which items are selected by the user, or to programmatically select items if needed.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.shadowColor Property

Gets or sets the color used to draw item's shadow.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get shadowColor() {}
set shadowColor(value) {}
```

Property Value

String

A string specifying the shadow color.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.shadowOffsetX Property

Gets or sets the horizontal offset of item's shadow.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get shadowOffsetX() {}
set shadowOffsetX(value) {}
```

Property Value

Number

A number specifying the shadow's horizontal offset.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.shadowOffsetY Property

Gets or sets the vertical offset of item's shadow.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get shadowOffsetY() {}
set shadowOffsetY(value) {}
```

Property Value

Number

A number specifying the shadow's vertical offset.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.stroke Property

Gets or sets the color used to stroke item's frame.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get stroke() {}
set stroke(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
brush Property
opacity Property

DiagramItem.strokeDashArray Property

Gets or sets an array of values specifying the dash pattern.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get strokeDashArray() {}
set strokeDashArray(value) {}
```

Property Value

Array

An array of numbers specifying the dash pattern.

Remarks

This value is used when strokeDashStyle is set to Custom.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.strokeDashOffset Property

Gets or sets the the distance between the line start and the beginning of the dash pattern.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get strokeDashOffset() {}
set strokeDashOffset(value) {}
```

Property Value

Number

The dash offset.

Remarks

This value is used when strokeDashStyle is set to Custom.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.strokeDashStyle Property

Gets or sets the line dash pattern applied when stroking item's frame.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get strokeDashStyle() {}
set strokeDashStyle(value) {}
```

Property Value

DashStyle

A member of the DashStyle enumeration.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.strokeThickness Property

Gets or sets the line width applied when stroking item's frame.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get strokeThickness() {}
set strokeThickness(value) {}
```

Property Value

Number

A number specifying the stroke thickness.

Remarks

Setting this value to zero makes stroke one-pixel wide regardless of diagram's measureUnit.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.style Property

Gets or sets the style associated with this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get style() {}
set style(value) {}
```

Property Value

Style

An instance of the Style class.

Remarks

Individual properties of DiagramItem have higher priority than ones set through a Style object. If neither are set, values are inherited from Diagram's style or from current theme.

Note

As of version 4.9 of the library, items can also by styled via CSS, where a custom CSS property name inside a declaration matches respective Style property, but is hyphenated, and type selectors match item type name with added mfd- prefix. For example:

```
mfd-shapenode {
   --brush: lightblue;
   --stroke: blue;
   --stroke-thickness: 2px;
   --font-name: 'Verdana';
   --font-size: 12px;
}
```

You can also associate CSS rules with individual items by setting cssClass property and specifying it in class selectors.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.tag Property

Gets or sets custom data tag associated with this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get tag() {}
set tag(value) {}
```

Property Value

Object

An Object containing custom data.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.text Property

Gets or sets the text label displayed by this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript — Copy Code

```
get text() {}
set text(value) {}
```

Property Value

String

A string containing the item's text. The default value is empty string "".

Remarks

This property defines item's main text content. You can add more text labels by calling the addLabel method, and customize their position and appearance via the various members of returned ItemLabel object.

Users can double-click an item to edit its text interactively when the allowInplaceEdit property is enabled. You can trigger the edit operation from a different event handler by calling beginEdit.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
font Property
textAlignment Property
lineAlignment Property
textColor Property

DiagramItem.textAlignment Property

Gets or sets the horizontal alignment of item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript — Copy Code

```
get textAlignment() {}
set textAlignment(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
text Property
lineAlignment Property

DiagramItem.textColor Property

Gets or sets the color used to fill item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

A string value identifying the text fill color.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
text Property
font Property

DiagramItem.textComponent Property

Gets the Text object instance used to render item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get textComponent() {}
set textComponent(value) {}
```

Property Value

Text

The Text component.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.textPadding Property

Gets or sets the padding distance between item's boundaries and text label.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get textPadding() {}
set textPadding(value) {}
```

Property Value

Thickness

An instance of the Thickness class.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
text Property

DiagramItem.textStroke Property

Gets or sets the color used to stroke item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get textStroke() {}
set textStroke(value) {}
```

Property Value

String

A string value identifying the text stroke color.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.textStrokeThickness Property

Gets or sets the outline width of item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get textStrokeThickness() {}
set textStrokeThickness(value) {}
```

Property Value

Number

A number specifying the thickness of text outline.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.tooltip Property

Gets or sets tooltip text that should be displayed when the mouse hovers over this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get tooltip() {}
set tooltip(value) {}
```

Property Value

String

A string containing the tooltip text.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.topLevel Property

Gets if this is a top-level item in the diagram.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get topLevel() {}
```

Property Value

Boolean

true if this is a top-level item in the diagram, or **false** if inside a container.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.visible Property

Gets or sets a value indicating whether this item is visible.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get visible() {}
set visible(value) {}
```

Property Value

Boolean

true if this item should be displayed, or **false** otherwise.

Remarks

Invisible items are not drawn at all. You might use the property to hide or show items temporarily, or to construct more complex groups of attached items in which intermediate helper items are invisible.

Invisible nodes are ignored by automatic layout classes. The position of such nodes won't change when the arrange method of Diagram is called. Visible items can be ignored too by enabling their ignoreLayout property.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.weight Property

Gets or sets a weight value used by layout and path-finding algorithms.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get weight() {}
set weight(value) {}
```

Property Value

Number

A number specifying the weight of this item. The default value is 1.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.zIndex Property

Gets the z-order position of this item.

Namespace: MindFusion.Diagramming
Assembly: DiagramItem.js

Syntax

JavaScript Copy Code

```
get zIndex() {}
set zIndex(value) {}
```

Property Value

Number

A number specifying the position of this item within diagram's z-order.

Remarks

Use the method to arrange items in depth. Items with lower **zIndex** appear behind items with higher **zIndex**.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramLink Class

Represents a link between two diagram nodes.

Syntax

JavaScript — Copy Code

```
// class
DiagramLink.prototype = {}
```

Remarks

Links between nodes in MvcDiagram are represented by instances of the **DiagramLink** class. Links connect at most two nodes, which can be accessed via the origin and destination properties. If connected to node anchor points, the point indices can be accessed using the originAnchor and destinationAnchor properties.

Links are built of smaller parts called segments, which can be straight lines, cascading orthogonal lines or Bezier curves. The type of segments is set through the shape property. The number of link's segments and their locations are specified by the segmentCount and controlPoints properties. If changing the coordinates of individual points in **controlPoints**, you should also call the updateFromPoints method to update link's internal state.

Typically, only the start and end segments of a link are updated when the nodes it connects are moved. Number, orientation and/or position of link segments can change dynamically in various ways depending on the values of autoRoute and dynamic properties.

Links can display distinct arrowhead shapes at their ends. These shapes are configured using the headShape and baseShape properties. The sizes of the base and arrowhead images are specified with the headShapeSize and baseShapeSize properties. Arrowhead shapes can also be displayed at the midpoint of each segment to help distinguish between different links. The type and size of these shapes are controlled by the intermediateShape and intermediateShapeSize properties. The outlines of the arrowhead shapes are rendered using the headStroke color.

The color of link segments is specified by the stroke property, while the fill color of the base and head shapes is controlled by the baseBrush and headBrush properties.

Links can have associated text, displayed according to the text, textColor, and textStyle properties. The addLabel method allows multiple labels to be displayed per link. Labels can automatically avoid overlapping nodes or other labels when their autoArrange property is enabled.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramLink

See Also

DiagramLink Members
MindFusion.Diagramming Namespace

DiagramLink Members

Represents a link between two diagram nodes.

The following tables list the members exposed by the DiagramLink type.

Public Constructors

	Name	Description
	DiagramLink	Initializes a new instance of the DiagramLink class between the specified nodes using the specified diagram link as a prototype.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	autoRoute	Gets or sets whether this link should be automatically routed to avoid crossing nodes.

	baseBrush	Gets or sets an object that specifies how to fill link's baseShape.
	baseShape	Gets or sets the arrowhead shape displayed at the link' start point.
	baseShapeSize	Gets or sets the size of the arrowhead displayed at the link's start point.
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	controlPoints	Gets or sets the control points of this link.
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	destination	Gets or sets the destination node of this link.
	destinationAnchor	Gets or sets the anchor point to which the link connects at its destination.
	destinationIndex	Gets or sets the index of link's destination row when destination is a TableNode.
	dynamic	Gets or sets whether link's end segments are automatically adjustedin order to keep pointing to the centers of the nodes it connects.
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	endPoint	Gets or sets the last control point of this link.
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	headBrush	Gets or sets an object that specifies how to fill link's headShape.
	headShape	Gets or sets the arrowhead shape displayed at the link' end point.
	headShapeSize	Gets or sets the size of the arrowhead displayed at the link's end point.
	headStroke	Gets or sets the brush used to stroke the outline of link's arrowheads.
	headStrokeDashStyle	Gets or sets the dash pattern of arrowhead outlines.
	headStrokeThickness	Gets or sets the thickness of arrowhead outlines.
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	intermediateBrush	Gets or sets an object that specifies how to fill intermediateShape arrowheads.
	intermediateShape	Gets or sets the arrowhead shape displayed at the middle of each segment of the link.
	intermediateShapeSize	Gets or sets the size of the arrowhead displayed at the middle of each segment of the link.
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)

	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	origin	Gets or sets the origin node of this link.
	originAnchor	Gets or sets the anchor point to which the link connects at its origin.
	originIndex	Gets or sets the index of link's origin row when origin is a **TableNode**.
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	segmentCount	Gets or sets the number of segments that form the link's geometry.
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	shape	Gets or sets the type of link segments and how they are oriented relatively to each other.
	startPoint	Gets or sets the first control point of this link.
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	text	Gets or sets the text displayed by this link.
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	textAlignment	Gets or sets the alignment of link's text.
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	textStyle	Gets or sets the placement and orientation of link's text.
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	True if this is a top-level item in the diagram, or false if it is inside a container. (Inherited from DiagramItem.)

	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	addLabel	Adds a new label to this link.
	allowDrag	DiagramItem.allowDrag override.
	Animate	Animates the item. (Inherited from DiagramItem.)
	clone	DiagramItem.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	containsPoint	DiagramItem.containsPoint override.
	ContainsPoint	Overloaded. Determines whether this item contains the specified point. (Inherited from DiagramItem.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	Draws the item on the specified IGraphics surface. (Inherited from DiagramItem.)
	DrawHandles	Renders the adjustment handles of this item. (Inherited from DiagramItem.)
	DrawInRect	(Inherited from DiagramItem.)
	DrawShadow	Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramItem.)
	endDrag	DiagramItem.endDrag override.
	fromJson	DiagramItem.fromJson override.
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	getLength	Gets the length of the links
	GetOutlinePoly	Gets a polygon approximating the outline of this object. (Inherited from DiagramItem.)

GetRepaintRect	Gets the repaint region for this item, taking into account factors such as pen, selection handles and shadow. (Inherited from DiagramItem.)
GetRotatedBounds	Gets the bounding rectangle of this item when it is rotated. (Inherited from DiagramItem.)
GetToolTip	Gets the tooltip that should be displayed for this item at the specified point. (Inherited from DiagramItem.)
HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
HitTestHandle	Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramItem.)
HitTestManipulators	Finds the **Manipulator** object located at the specified point. (Inherited from DiagramItem.)
intersects	Checks whether the link contour intersects with the outline of the specified node.
LabelFromPoint	Returns the ItemLabel located at specified point. (Inherited from DiagramItem.)
loadFromXml	DiagramItem.LoadFromXml override. Loads the link's content from an XML element.
NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
pointAlongLength	Returns a point lying on a polyline that is positioned at a relative position along the polyline's length.
putEndPointsAtNodeBorders	Places the link's end points on the outline of its origin and destination nodes.
Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
removeLabel	Removes the specified label from this link.
RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
restoreLocationState	DiagramItem.restoreLocationState override.
restoreState	DiagramItem.restoreState override.
route	Routes this link, regardless of whether the link's **autoRoute** property is enabled.
SaveItemProperties	For internal use. (Inherited from DiagramItem.)
saveLocationState	**DiagramItem.saveLocationState** override.
saveState	DiagramItem.saveState override.
saveToXml	Saves the item's content into an XML element.
setDestination	Sets the destination node of this link.
splitSegment	Splits the link segment with the specified index at the specified point.
startDrag	DiagramItem.startDrag override.
toJson	DiagramItem.toJson override.
updateCanvasElements	Updates the elements on the HTML Canvas.
updateDrag	DiagramItem.updateDrag override.

	updateFromPoints	Updates the link's internal state after the link's control points have been changed.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	Invoked during interactive creation of items to test whether the operation can be completed in the current context. (Inherited from DiagramItem.)
	AllowModify	Invoked during interactive modification of items to test whether the operation can be completed in the current context. (Inherited from DiagramItem.)
	CancelModify	Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramItem.)
	CompleteCreate	Invoked when the item creation has been completed. (Inherited from DiagramItem.)
	CompleteModify	Invoked when the item modification has been completed. (Inherited from DiagramItem.)
	CreateProperties	Creates an item property set corresponding to the type of this diagram item. (Inherited from DiagramItem.)
	CreateState	Creates an item state corresponding to the type of this diagram item. (Inherited from DiagramItem.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the item is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramItem.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	LoadFromJson	Loads the item from a JSON object. (Inherited from DiagramItem.)
	LoadFromXml	Loads the item content from an XML element. (Inherited from DiagramItem.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	Called when the item is added to a Diagram, either by the user, programmatically or via undo / redo. (Inherited from DiagramItem.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)

	OnMeasureUnitChanged	Called when the diagram's MeasureUnit changes. (Inherited from DiagramItem.)
	OnRemove	Called when an item is removed from a **Diagram**, either by the user, programmatically or via undo / redo. (Inherited from DiagramItem.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnZoomChanged	Called when the diagram's **ZoomFactor** changes. (Inherited from DiagramItem.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	Restores the properties of this diagram item. (Inherited from DiagramItem.)
	RestoreState	Restores the state of this diagram item. (Inherited from DiagramItem.)
	SaveProperties	Saves the properties of this diagram item. (Inherited from DiagramItem.)
	SaveState	Saves the state of this diagram item. (Inherited from DiagramItem.)
	SaveToJson	Saves the item as a JSON object. (Inherited from DiagramItem.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramItem.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpanded	Sets whether the item is in expanded or collapsed state. (Inherited from DiagramItem.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	StartCreate	Invoked once when the user starts creating a new item interactively. (Inherited from DiagramItem.)
	StartModify	Invoked when the user starts modifying this item interactively. (Inherited from DiagramItem.)
	UpdateCreate	Invoked during interactive creation of this item. (Inherited from DiagramItem.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateModify	Invoked during interactive modification of this item. (Inherited from DiagramItem.)

See Also

DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Constructor

Initializes a new instance of the DiagramLink class between the specified nodes using the specified diagram link as a prototype.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function DiagramLink ([parent, [origin, [destination]]])
```

Parameters

parent
Optional.
Diagram. The Diagram from which to obtain default values for the link properties.

origin
Optional.
DiagramNode | Point. The origin node of the new link.

destination
Optional.
DiagramNode | Point. The destination node of the new link.

See Also

DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Methods

DiagramLink.addLabel Method

Adds a new label to this link.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function addLabel (text)
```

Parameters

text
String. The label's text.

Return Value

A LinkLabel instance.

Remarks

A link's main text content is specified via its text property. Call **addLabel** if you need to display additional text labels, such as identifiers or hints to the user.

Position of labels can be specified by calling methods such as setControlPointPosition. You can let users move labels interactively by setting the behavior property of DiagramView to MoveLabels.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
removeLabel Method

DiagramLink.allowDrag Method

DiagramItem.allowDrag override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function allowDrag (ist)
```

Parameters

ist
An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.clone Method

DiagramItem.clone override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function clone ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.containsPoint Method

DiagramItem.containsPoint override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function containsPoint (point)
```

Parameters

point
The Point to check.

Return Value

Boolean. **true** if the link contains the point, or **false** otherwise.

See Also

DiagramLink Members
DiagramLink Class

DiagramLink.endDrag Method

DiagramItem.endDrag override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function endDrag (ist)
```

Parameters

ist
An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.fromJson Method

DiagramItem.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function fromJson ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.getLength Method

Gets the length of the links

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function getLength ([segments])
```

Parameters

segments
Optional.
Array. A list of points specifying the polyline segments.

Return Value

Number. The sum of the lengths of the line's segments.

Remarks

If the link's Shape is set to Bezier, an approximate length is calculated.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.intersects Method

Checks whether the link contour intersects with the outline of the specified node.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function intersects (rect)
```

Parameters

rect
Rect. The bounding rectangle of the specified node.

Return Value

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.loadFromXml Method

DiagramItem.LoadFromXml override. Loads the link's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.pointAlongLength Method

Returns a point lying on a polyline that is positioned at a relative position along the polyline's length.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript — Copy Code

```
function pointAlongLength (lengthFactor, [segments])
```

Parameters

lengthFactor
Number. A value from 0 to 1 specifying a portion of link's length.

segments
Optional.
Array. A list of points specifying the polyline segments.

Return Value

Point. A Point instance.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.putEndPointsAtNodeBorders Method

Places the link's end points on the outline of its origin and destination nodes.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript — Copy Code

```
function putEndPointsAtNodeBorders (action, routing)
```

Parameters

action

routing

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.removeLabel Method

Removes the specified label from this link.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript — Copy Code

```
function removeLabel (label)
```

Parameters

label
A LinkLabel instance.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
addLabel Method

DiagramLink.restoreLocationState Method

DiagramItem.restoreLocationState override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function restoreLocationState (ist)
```

Parameters

ist
A **DiagramLinkState** instance.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.restoreState Method

DiagramItem.restoreState override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function restoreState ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.route Method

Routes this link, regardless of whether the link's autoRoute property is enabled.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function route ([args])
```

Parameters

args
Optional.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.saveLocationState Method

DiagramItem.saveLocationState override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function saveLocationState ()
```

Return Value

DiagramLinkState. A DiagramLinkState instance.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.saveState Method

DiagramItem.saveState override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function saveState ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.saveToXml Method

Saves the item's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.setDestination Method

Sets the destination node of this link.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function setDestination (destination)
```

Parameters

destination
The DiagramNode to set as destination for this link.

Remarks

For internal use.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
origin Property
destination Property

DiagramLink.splitSegment Method

Splits the link segment with the specified index at the specified point.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function splitSegment ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.startDrag Method

DiagramItem.startDrag override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function startDrag (ist)
```

Parameters

ist
An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.toJson Method

DiagramItem.toJson override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.updateCanvasElements Method

Updates the elements on the HTML Canvas.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function updateCanvasElements ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.updateDrag Method

DiagramItem.updateDrag override.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function updateDrag (ist)
```

Parameters

ist
An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.updateFromPoints Method

Updates the link's internal state after the link's control points have been changed.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
function updateFromPoints ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Properties

DiagramLink.autoRoute Property

Gets or sets whether this link should be automatically routed to avoid crossing nodes.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get autoRoute() {}
set autoRoute(value) {}
```

Property Value

Boolean

true to automatically route this link, or **false** otherwise.

Remarks

If **autoRoute** is enabled, a link automatically finds the shortest possible path between its origin and destination that doesn't intersect obstacle nodes. The number and orientation of the link's segments are updated when the nodes it links are moved around. The positions of end points of a routed link are not changed, unless the link is also set to dynamic.

> **Performance Note**
>
> Auto-routing is CPU intensive operation and is triggered each time a node is moved or resized. If populating large diagrams from data, best enable this property only after creating all nodes. Alternatively, call route or routeAllLinks method explicitly when appropriate for your application.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
dynamic Property

DiagramLink.baseBrush Property

Gets or sets an object that specifies how to fill link's baseShape.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get baseBrush() {}
set baseBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

Remarks

If this property is not set, the arrowhead is filled using the link's brush.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.baseShape Property

Gets or sets the arrowhead shape displayed at the link' start point.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get baseShape() {}
set baseShape(value) {}
```

Property Value

Shape | *String*

A Shape instance that represents an arrowhead shape.

Remarks

This property specifies the shape that is displayed at the beginning of a link, that is, the point where it connects to its origin node. This shape is rotated simultaneously with the first segment of the link so that it follows the overall link direction.

The ArrowHeads class exposes predefined arrowhead shapes as static methods, each one returning Shape.fromId result for respective **shape identifier**.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
headShape Property
intermediateShape Property

DiagramLink.baseShapeSize Property

Gets or sets the size of the arrowhead displayed at the link's start point.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get baseShapeSize() {}
set baseShapeSize(value) {}
```

Property Value

Number

A number that represents the size of the arrowhead shape.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.controlPoints Property

Gets or sets the control points of this link.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get controlPoints() {}
set controlPoints(value) {}
```

Property Value

Array

An array of Point objects containing coordinates of link segments.

Remarks

This property lets you change coordinates of link points. The number of points per segment depends on link's shape. Links made of Bézier-splines have four control points for each segment. Links with other shape types have two control points per segment. Adjacent segments share a common control point. After setting new positions of control points, you must call the updateFromPoints method.

See Also

DiagramLink.destination Property

Gets or sets the destination node of this link.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get destination() {}
set destination(value) {}
```

Property Value

DiagramNode

A reference to the destination DiagramNode.

Remarks

When handling validation events, this property keeps pointing to the original destination node. Use the destination property of LinkEventArgs to determine which node the link will connect to if operation is approved.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
origin Property
destinationIndex Property

DiagramLink.destinationAnchor Property

Gets or sets the anchor point to which the link connects at its destination.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get destinationAnchor() {}
set destinationAnchor(value) {}
```

Property Value

Number

An integer value specifying the index of the destination anchor point.

Remarks

This property is valid when destination node's anchorPattern is set.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.destinationIndex Property

Gets or sets the index of link's destination row when destination is a TableNode.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get destinationIndex() {}
set destinationIndex(value) {}
```

Property Value

Number

An integer value specifying the index of destination row.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.dynamic Property

Gets or sets whether link's end segments are automatically adjustedin order to keep pointing to the centers of the nodes it connects.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get dynamic() {}
set dynamic(value) {}
```

Property Value

Boolean

true if the link automatically adjusts its end points, or false otherwise.

Remarks

If this property is enabled, moving link's adjacent nodes or next-to-last link control points will also update the position of link end points along the node border.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
autoRoute Property

DiagramLink.endPoint Property

Gets or sets the last control point of this link.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get endPoint() {}
set endPoint(value) {}
```

Property Value

Point

A Point instance containing the control point coordinates.

Remarks

This is a shortcut for accessing last element of controlPoints.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
startPoint Property

DiagramLink.headBrush Property

Gets or sets an object that specifies how to fill link's headShape.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get headBrush() {}
set headBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

Remarks

If this property is not set, the arrowhead is filled using the link's brush.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.headShape Property

Gets or sets the arrowhead shape displayed at the link' end point.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get headShape() {}
set headShape(value) {}
```

Property Value

Shape | *String*

A Shape instance that represents an arrowhead shape.

Remarks

This property specifies the shape that is displayed at the end of a link, that is, the point where it connects to its destination node. This shape is rotated simultaneously with the last segment of the link so that it follows the overall link direction.

The ArrowHeads class exposes predefined arrowhead shapes as static methods, each one returning Shape.fromId result for respective **shape identifier**.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
baseShape Property
intermediateShape Property

DiagramLink.headShapeSize Property

Gets or sets the size of the arrowhead displayed at the link's end point.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get headShapeSize() {}
set headShapeSize(value) {}
```

Property Value

Number

A number that represents the size of the arrowhead shape.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
headShape Property

DiagramLink.headStroke Property

Gets or sets the brush used to stroke the outline of link's arrowheads.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get headStroke() {}
set headStroke(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
headShape Property
baseShape Property

DiagramLink.headStrokeDashStyle Property

Gets or sets the dash pattern of arrowhead outlines.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get headStrokeDashStyle() {}
set headStrokeDashStyle(value) {}
```

Property Value

DashStyle

A member of the DashStyle enumeration.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.headStrokeThickness Property

Gets or sets the thickness of arrowhead outlines.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get headStrokeThickness() {}
set headStrokeThickness(value) {}
```

Property Value

Number

A number specifying the line thickness.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.intermediateBrush Property

Gets or sets an object that specifies how to fill intermediateShape arrowheads.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get intermediateBrush() {}
set intermediateBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

Remarks

If this property is not set, intermediate arrowheads are filled using the link's brush.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
intermediateShapeSize Property

DiagramLink.intermediateShape Property

Gets or sets the arrowhead shape displayed at the middle of each segment of the link.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get intermediateShape() {}
set intermediateShape(value) {}
```

Property Value

Shape | *String*

A Shape instance that represents an arrowhead shape.

Remarks

The property specifies what shape to display at the middle of each segment of a link. This could help users more easily follow the path of long links.

The ArrowHeads class exposes predefined arrowhead shapes as static methods, each one returning Shape.fromId result for respective shape identifier.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
headShape Property
baseShape Property

DiagramLink.intermediateShapeSize Property

Gets or sets the size of the arrowhead displayed at the middle of each segment of the link.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get intermediateShapeSize() {}
set intermediateShapeSize(value) {}
```

Property Value

Number

A number that represents the size of the arrowhead shape.

See Also

DiagramLink.origin Property

Gets or sets the origin node of this link.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get origin() {}
set origin(value) {}
```

Property Value

DiagramNode

A reference to the origin DiagramNode.

Remarks

When handling validation events, this property keeps pointing to the original origin node. Use the origin property of LinkEventArgs to determine which node the link will connect to if operation is approved.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
destination Property
originIndex Property

DiagramLink.originAnchor Property

Gets or sets the anchor point to which the link connects at its origin.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get originAnchor() {}
set originAnchor(value) {}
```

Property Value

Number

An integer value specifying the index of the origin anchor point.

Remarks

This property is valid when origin node's anchorPattern is set.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.originIndex Property

Gets or sets the index of link's origin row when origin is a TableNode.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get originIndex() {}
set originIndex(value) {}
```

Property Value

Number

An integer value specifying the index of origin row.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.segmentCount Property

Gets or sets the number of segments that form the link's geometry.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get segmentCount() {}
set segmentCount(value) {}
```

Property Value

Number

An integer value specifying the number of segments of this link.

Remarks

Link segments are either straight lines or Bezier curves, depending on the value of the link's shape property. The segment coordinates are defined by the elements of controlPoints, which contains four points per Bezier segment or two points per straight line segment, where each pair of adjacent segments share a common point. Changing the **segmentCount** property value will resize the **controlPoints** array so that it contains appropriate number of points.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.shape Property

Gets or sets the type of link segments and how they are oriented relatively to each other.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get shape() {}
set shape(value) {}
```

Property Value

LinkShape

A member of the LinkShape enumeration.

Remarks

This property specifies whether the segments of a link are straight lines or curves. In case of straight lines, they can be set all horizontal and vertical, each segment orthogonal to its neighbor, by using the Cascading shape.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
controlPoints Property

DiagramLink.startPoint Property

Gets or sets the first control point of this link.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get startPoint() {}
set startPoint(value) {}
```

Property Value

Point

A Point instance containing the control point coordinates.

Remarks

This is a shortcut for accessing first element of controlPoints.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
endPoint Property

DiagramLink.text Property

Gets or sets the text displayed by this link.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get text() {}
set text(value) {}
```

Property Value

String

A string containing the link's text. The default value is empty string "".

Remarks

Gets or sets the text that should be drawn over a link. Use this to provide to end-users a short description of the relation's purpose. The text appearance can be customized through the textColor and textStyle properties.

Version 1.6 of the control adds the capability to display multiple labels per link via the LinkLabel class. Labels can automatically avoid overlapping nodes or other labels when their autoArrange property is set.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.textAlignment Property

Gets or sets the alignment of link's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get textAlignment() {}
set textAlignment(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration. The default value is **Center**.

Remarks

Currently this property is considered only when textStyle is set to OverLongestSegment.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.textStyle Property

Gets or sets the placement and orientation of link's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramLink.js

Syntax

JavaScript Copy Code

```
get textStyle() {}
set textStyle(value) {}
```

Property Value

LinkTextStyle

A member of the LinkTextStyle enumeration.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramNode Class

Represents a diagram node.

Syntax

JavaScript Copy Code

```
// class
DiagramNode.prototype = {}
```

Remarks

DiagramNode objects can represent graph vertices, nodes in organizational or flow diagrams, entities in ER diagrams, and so on. **DiagramNode** is an abstract class. You cannot instantiate it directly, but instead use objects from specific classes such as ShapeNode, TableNode, ContainerNode, CompositeNode. You can also define your own node type by deriving from **DiagramNode** or its descendants.

Use the bounds property to get or change the location and size of a node. Constraints upon how users are allowed to modify a node are set through the constraints and enabledHandles properties. The former specifies constraints on move direction and node size. The latter enables or disables specific manipulation handles of a node.

The main textual content of a node is set using its text property and can be customized by setting font and textAlignment. Supplementary labels can be created by calling addLabel.

Relations between nodes are indicated by connecting the nodes with DiagramLink instances. Links are considered directed graph edges, so each node has two collections referring to links: incomingLinks and outgoingLinks. anchorPattern lets you specify the points of a node to which incoming or outgoing links can be connected. The allowIncomingLinks property lets you specify whether incoming links can be drawn to a node. The allowOutgoingLinks property lets you do that for outgoing links.

A user is allowed to expand or collapse the tree starting from a node if expandable is set to **true**. The expanded property gets or sets whether the subtree starting from a node is expanded or not.

Nodes can be attached to other items, and be used as simple labels, image-holders or other kinds of UI elements. The attachTo method overloads provide various options for that.

Nodes can be rotated, either interactively using the rotation handle above a node, or programmatically by setting the rotationAngle property.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode

See Also

DiagramNode Members
MindFusion.Diagramming Namespace

DiagramNode Members

Represents a diagram node.

The following tables list the members exposed by the DiagramNode type.

Public Constructors

	Name	Description
	DiagramNode	Initializes a new instance of the **DiagramNode** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description

	allowIncomingLinks	Gets or sets whether users are allowed to draw incoming links to this node.
	allowOutgoingLinks	Gets or sets whether users are allowed to draw outgoing links from this node.
	anchorPattern	Gets or sets the anchor points to which links attach when connected to this node.
	attachedNodes	Returns the nodes attached to this node.
	bounds	Gets or sets a rectangle that defines the position of this node.
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	center	Gets the center of this node's bounding rectangle.
	constraints	Gets or sets constraints applied to the location and size of this node.
	cornerRadius	Gets or sets the corner radius of nodes whose shape is set to a rounded rectangle.
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	effectiveEffects	Gets the effects applied to this node.
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	effects	Gets the effects applied to this node.
	enabledHandles	Gets or sets a value indicating which adjustment handles are enabled for this node.
	expandable	Gets or sets whether users are allowed to expand or collapse the subtree rooted at this node.
	expanded	Gets or sets whether the subtree rooted at this node is expanded.
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	handlesStyle	Gets or sets a value specifying appearance and behavior of node's adjustment handles.
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	incomingLinks	Gets the incoming links to this node.
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)

	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	masterNode	Returns the node to which this node is attached.
	obstacle	Gets or sets whether this node is considered an obstacle by link routing algorithms.
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	outgoingLinks	Gets the outgoing links from this node.
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	rotationAngle	Gets or sets the angle at which this node is rotated.
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	showDeleteButton	Gets or sets whether this node displays a delete button.
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	True if this is a top-level item in the diagram, or false if it is inside a container. (Inherited from DiagramItem.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	acceptLinks	Determines whether the node can be connected with links of the specified type.
	allowDrag	DiagramItem.allowDrag override.
	Animate	Animates the item. (Inherited from DiagramItem.)
	attach	Attaches the specified subordinate node to this node, so that when this node is moved, the subordinate follows it.
	attachTo	Attaches this node to the specified master node so that when the master is moved, the attached node follows it.
	clone	DiagramItem.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	containsPoint	DiagramItem.containsPoint override.
	ContainsPoint	Overloaded. Determines whether this item contains the specified point. (Inherited from DiagramItem.)
	createConnectionPoint	Creates a connection point between this node and the specified link at the specified point.
	detach	Detaches this node from its current master node.
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	Draws the item on the specified IGraphics surface. (Inherited from DiagramItem.)
	DrawHandles	Renders the adjustment handles of this item. (Inherited from DiagramItem.)
	DrawInRect	(Inherited from DiagramItem.)
	DrawShadow	Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramItem.)
	endDrag	**DiagramItem.endDrag** override.
	fromJson	DiagramItem.fromJson override.
	getAllLinks	Returns all links connected to this node.
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)

	getIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline.
	getOutline	Gets the bounding rectangle of this node.
	GetOutlinePoly	Gets a polygon approximating the outline of this object. (Inherited from DiagramItem.)
	GetRepaintRect	Gets the repaint region for this item, taking into account factors such as pen, selection handles and shadow. (Inherited from DiagramItem.)
	getRotatedBounds	Gets the bounding **rectangle** of this item when it is rotated.
	GetRotatedBounds	Gets the bounding rectangle of this item when it is rotated. (Inherited from DiagramItem.)
	GetToolTip	Gets the tooltip that should be displayed for this item at the specified point. (Inherited from DiagramItem.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramItem.)
	HitTestManipulators	Finds the **Manipulator** object located at the specified point. (Inherited from DiagramItem.)
	LabelFromPoint	Returns the ItemLabel located at specified point. (Inherited from DiagramItem.)
	loadFromXml	DiagramItem.loadFromXml override. Loads the node's content from an XML element.
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	raiseClicked	Raises an event which notifies that a **node** has been clicked.
	raiseDoubleClicked	Raises an event which notifies that a node has been double-clicked.
	removeCanvasElements	DiagramItem.removeCanvasElements override.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	restoreLocationState	DiagramItem.restoreLocationState override.
	restoreState	DiagramItem.restoreState override.
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveLocationState	**DiagramItem.saveLocationState** override.
	saveState	DiagramItem.saveState override.
	saveToXml	Saves the item content into an XML element.
	setBounds	Sets the rectangle that defines the position of the diagram node.
	toJson	DiagramItem.toJson override.
	updateDrag	**DiagramItem.updateDrag** override.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)

	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	Invoked during interactive creation of items to test whether the operation can be completed in the current context. (Inherited from DiagramItem.)
	AllowModify	Invoked during interactive modification of items to test whether the operation can be completed in the current context. (Inherited from DiagramItem.)
	CancelModify	Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramItem.)
	CompleteCreate	Invoked when the item creation has been completed. (Inherited from DiagramItem.)
	CompleteModify	Invoked when the item modification has been completed. (Inherited from DiagramItem.)
	CreateProperties	Creates an item property set corresponding to the type of this diagram item. (Inherited from DiagramItem.)
	CreateState	Creates an item state corresponding to the type of this diagram item. (Inherited from DiagramItem.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the item is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramItem.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	LoadFromJson	Loads the item from a JSON object. (Inherited from DiagramItem.)
	LoadFromXml	Loads the item content from an XML element. (Inherited from DiagramItem.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	Called when the item is added to a Diagram, either by the user, programmatically or via undo / redo. (Inherited from DiagramItem.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	Called when the diagram's MeasureUnit changes. (Inherited from DiagramItem.)
	OnRemove	Called when an item is removed from a **Diagram**, either by the user, programmatically or via undo / redo. (Inherited from DiagramItem.)

	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnZoomChanged	Called when the diagram's **ZoomFactor** changes. (Inherited from DiagramItem.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	Restores the properties of this diagram item. (Inherited from DiagramItem.)
	RestoreState	Restores the state of this diagram item. (Inherited from DiagramItem.)
	SaveProperties	Saves the properties of this diagram item. (Inherited from DiagramItem.)
	SaveState	Saves the state of this diagram item. (Inherited from DiagramItem.)
	SaveToJson	Saves the item as a JSON object. (Inherited from DiagramItem.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramItem.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpanded	Sets whether the item is in expanded or collapsed state. (Inherited from DiagramItem.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	StartCreate	Invoked once when the user starts creating a new item interactively. (Inherited from DiagramItem.)
	StartModify	Invoked when the user starts modifying this item interactively. (Inherited from DiagramItem.)
	UpdateCreate	Invoked during interactive creation of this item. (Inherited from DiagramItem.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateModify	Invoked during interactive modification of this item. (Inherited from DiagramItem.)

See Also

DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode Constructor

Initializes a new instance of the DiagramNode class.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function DiagramNode ([parent])
```

Parameters

parent
Optional.
Diagram. A Diagram instance whose default node attributes are copied to this node.

See Also

DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode Methods

DiagramNode.acceptLinks Method

Determines whether the node can be connected with links of the specified type.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function acceptLinks (outgoing)
```

Parameters

outgoing
Boolean. **true** to check for outgoing links, or **false** otherwise.

Return Value

Boolean. **true** if the node accepts connections of the specified type, or **false** otherwise.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.allowDrag Method

DiagramItem.allowDrag override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function allowDrag (ist)
```

Parameters

ist
InteractionState. An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.attach Method

Attaches the specified subordinate node to this node, so that when this node is moved, the subordinate follows it.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript — Copy Code

```
function attach (subordinate)
```

Parameters

subordinate
A DiagramNode instance specifying the node that should be attached.

Remarks

Layout algorithms can treat a group of attached nodes either as a single node to be placed in the arranged graph, or as distinct nodes whose final positions are set independently from the initial group layout. This is controlled via the keepGroupLayout property.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
detach Method
attachTo Method
attachedNodes Property

DiagramNode.attachTo Method

Attaches this node to the specified master node so that when the master is moved, the attached node follows it.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript — Copy Code

```
function attachTo (node)
```

Parameters

node
A DiagramNode instance specifying the master node.

Remarks

Layout algorithms can treat a group of attached nodes either as a single node to be placed in the arranged graph, or as distinct nodes whose final positions are set independently from the initial group layout. This is controlled via the keepGroupLayout property.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
detach Method
attach Method
attachedNodes Property

DiagramNode.clone Method

DiagramItem.clone override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function clone ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.containsPoint Method

DiagramItem.containsPoint override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function containsPoint (point)
```

Parameters

point
Point. The point to check.

Return Value

Boolean. true it this node contains the specified point; otherwise, false.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.createConnectionPoint Method

Creates a connection point between this node and the specified link at the specified point.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function createConnectionPoint (link, point, incoming)
```

Parameters

link
A DiagramLink that should be connected to this node.

point
The Point at which the link should be connected.

incoming

Boolean. **true** if link is an incoming link, or **false** otherwise.

Return Value

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.detach Method

Detaches this node from its current master node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function detach ()
```

Remarks

This method does nothing if the node hasn't been previously attached to another node via the attach or attachTo methods.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
attachedNodes Property

DiagramNode.endDrag Method

DiagramItem.endDrag override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function endDrag (ist)
```

Parameters

ist
An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.fromJson Method

DiagramItem.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function fromJson ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.getAllLinks Method

Returns all links connected to this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function getAllLinks ()
```

Return Value

Array. An array of DiagramLink objects.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.getIntersection Method

Calculates the intersection point between the segment defined by the specified point pair and the node's outline.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function getIntersection (point1, point2)
```

Parameters

point1

The first Point of the segment.

point2

The second **Point** of the segment.

Return Value

A **Point** object representing the intersection point.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.getOutline Method

Gets the bounding rectangle of this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript — Copy Code

```
function getOutline ()
```

Return Value

A rectangle.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.getRotatedBounds Method

Gets the bounding rectangle of this item when it is rotated.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript — Copy Code

```
function getRotatedBounds ()
```

Return Value

A Rect that bounds the rotated shape of this node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
rotationAngle Property

DiagramNode.loadFromXml Method

DiagramItem.loadFromXml override. Loads the node's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript — Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.raiseClicked Method

Raises an event which notifies that a node has been clicked.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function raiseClicked (mousePosition, mouseButton)
```

Parameters

mousePosition
Point. The position of the mouse.

mouseButton
A value identifying the clicked mouse button.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.raiseDoubleClicked Method

Raises an event which notifies that a node has been double-clicked.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function raiseDoubleClicked (mousePosition, mouseButton)
```

Parameters

mousePosition
Point. The position of the mouse.

mouseButton
MouseButton. A value identifying the clicked mouse button.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.removeCanvasElements Method

DiagramItem.removeCanvasElements override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function removeCanvasElements ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.restoreLocationState Method

DiagramItem.restoreLocationState override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function restoreLocationState (ist)
```

Parameters

ist
DiagramNodeState. A DiagramNodeState instance.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.restoreState Method

DiagramItem.restoreState override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function restoreState ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.saveLocationState Method

DiagramItem.saveLocationState override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function saveLocationState ()
```

Return Value

DiagramNodeState. A DiagramNodeState instance.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.saveState Method

DiagramItem.saveState override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function saveState ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.saveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement

Element. An XML DOM element that will contain the item's data.

context

XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.setBounds Method

Sets the rectangle that defines the position of the diagram node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function setBounds (bounds, [updateDependencies])
```

Parameters

bounds
The node's bounding rectangle.

updateDependencies
Optional. **true** if links and attachments must be updated too, or **false** otherwise.

Remarks

Use this method to set the size and/or location of a node. If updateDependencies is set, moving the node will also update the control-point coordinates of all links connected to it.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
bounds Property

DiagramNode.toJson Method

DiagramItem.toJson override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.updateDrag Method

DiagramItem.updateDrag override.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
function updateDrag (ist)
```

Parameters

ist
An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode Properties

DiagramNode.allowIncomingLinks Property

Gets or sets whether users are allowed to draw incoming links to this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get allowIncomingLinks() {}
set allowIncomingLinks(value) {}
```

Property Value

Boolean

true if users are allowed to draw incoming links, or **false** otherwise.

Remarks

Links to a node can always be created programmatically regardless of the value of this property.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.allowOutgoingLinks Property

Gets or sets whether users are allowed to draw outgoing links from this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get allowOutgoingLinks() {}
set allowOutgoingLinks(value) {}
```

Property Value

Boolean

true if users are allowed to draw outgoing links, or **false** otherwise.

Remarks

Links from a node can always be created programmatically regardless of the value of this property.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.anchorPattern Property

Gets or sets the anchor points to which links attach when connected to this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript — Copy Code

```
get anchorPattern() {}
set anchorPattern(value) {}
```

Property Value

AnchorPattern

An AnchorPattern object that defines a set of anchor points.

Remarks

This property specifies a set of points to which links can be attached. Each anchor point can accept either incoming links, outgoing links or both. If a node does not have any anchor points defined, links can be connected to any part of the node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
AnchorPoint Class
Shape.anchorPattern Property

DiagramNode.attachedNodes Property

Returns the nodes attached to this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript — Copy Code

```
get attachedNodes() {}
```

Property Value

Array

An array of nodes attached to this node via the attach or attachTo methods.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
detach Method

DiagramNode.bounds Property

Gets or sets a rectangle that defines the position of this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript — Copy Code

```
get bounds() {}
set bounds(value) {}
```

Property Value

Rect

The node's bounding rectangle.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Diagram.enableSpatialIndex Property

DiagramNode.center Property

Gets the center of this node's bounding rectangle.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get center() {}
```

Property Value

Point

A Point representing the center point of this node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.constraints Property

Gets or sets constraints applied to the location and size of this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get constraints() {}
set constraints(value) {}
```

Property Value

NodeConstraints

An instance of the NodeConstraints class.

Remarks

These constraints are considered when a user modifies the node interactively. They are ignored if node placement is changed programmatically by using methods and properties.

See Also

DiagramNode Members
DiagramNode Class

DiagramNode.cornerRadius Property

Gets or sets the corner radius of nodes whose shape is set to a rounded rectangle.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get cornerRadius() {}
set cornerRadius(value) {}
```

Property Value

Number

A number specifying the corner radius.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
ShapeNode.shape Property
TableNode.shape Property
ContainerNode.shape Property

DiagramNode.effectiveEffects Property

Gets the effects applied to this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get effectiveEffects() {}
```

Property Value

Array

An array of NodeEffect objects.

Remarks

Returns the local effects array if it's not empty, or otherwise the effects defined in item's style or diagram's theme.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.effects Property

Gets the effects applied to this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get effects() {}
set effects(value) {}
```

Property Value

Array

An array of NodeEffect objects.

Remarks

The local node effects are applied after the diagram-wide node effects defined in nodeEffects. Setting **null** marker as first element in this list allows ignoring the global diagram effects.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Node Effects

DiagramNode.enabledHandles Property

Gets or sets a value indicating which adjustment handles are enabled for this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get enabledHandles() {}
set enabledHandles(value) {}
```

Property Value

AdjustmentHandles

A combination of the AdjustmentHandles enumeration members. The default is **AdjustmentHandles.All**.

Remarks

Disabling some selection handles allows restricting the types of modifications that users can perform on a node. Each manipulation handle can be enabled or disabled by setting or clearing the respective flag in the combination.

For example, to make a node resizable only horizontally, set this property to a combination of the Move, ResizeMiddleLeft and ResizeMiddleRight flags. That will disable all manipulation handles, except the left and right ones, which allow horizontal resizing, and the central one allowing movement of the node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
handlesStyle Property
constraints Property

DiagramNode.expandable Property

Gets or sets whether users are allowed to expand or collapse the subtree rooted at this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get expandable() {}
set expandable(value) {}
```

Property Value

Boolean

true if users are allowed to expand/collapse the subtree, or **false** otherwise.

Remarks

This property is useful mostly when nodes form a tree-like hierarchical structure.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
expanded Property
treeCollapsed Event
treeExpanded Event
expandButtonAction Property

DiagramNode.expanded Property

Gets or sets whether the subtree rooted at this node is expanded.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get expanded() {}
set expanded(value) {}
```

Property Value

Boolean

true if the subtree is expanded, or **false** otherwise.

Remarks

By default, the control considers nodes that can be reached by following outgoing links to be the child nodes in a tree, and respectively outgoing links determine the direction in which tree branches collapse or expand. This direction can be reversed by setting the expandOnIncoming property to **true**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
expandable Property
outgoingLinks Property
incomingLinks Property

DiagramNode.handlesStyle Property

Gets or sets a value specifying appearance and behavior of node's adjustment handles.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get handlesStyle() {}
set handlesStyle(value) {}
```

Property Value

HandlesStyle

A member of the HandlesStyle enumeration. The default value is **SquareHandles**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
enabledHandles Property

DiagramNode.incomingLinks Property

Gets the incoming links to this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get incomingLinks() {}
```

Property Value

Array

An array of DiagramLink objects.

Remarks

Set link's destination property to programmatically create incoming link to a node, or set respective argument of createDiagramLink method.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.masterNode Property

Returns the node to which this node is attached.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get masterNode() {}
```

Property Value

DiagramNode

A DiagramNode reference specifying the node to whichthis node has been previously attached via the attach or attachTo method.

See Also

DiagramNode.obstacle Property

Gets or sets whether this node is considered an obstacle by link routing algorithms.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get obstacle() {}
set obstacle(value) {}
```

Property Value

Boolean

true to route links around this node, or **false** to lets links cross the node. The default value is **true**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.outgoingLinks Property

Gets the outgoing links from this node.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get outgoingLinks() {}
```

Property Value

Array

An array of DiagramLink objects.

Remarks

Set link's origin property to programmatically create outgoing link from a node, or set respective argument of createDiagramLink method.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.rotationAngle Property

Gets or sets the angle at which this node is rotated.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get rotationAngle() {}
set rotationAngle(value) {}
```

Property Value

Number

A number specifying the rotation angle. The default is 0.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.showDeleteButton Property

Gets or sets whether this node displays a delete button.

Namespace: MindFusion.Diagramming
Assembly: DiagramNode.js

Syntax

JavaScript Copy Code

```
get showDeleteButton() {}
set showDeleteButton(value) {}
```

Property Value

Boolean

true to display delete button for the node, or **false** otherwise.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramPage Class

Represents a single page in a DiagramDocument object.

Syntax

JavaScript Copy Code

```
// class
DiagramPage.prototype = {}
```

Remarks

DiagramPage is derived from Diagram and only adds a few properties, so any code that processes **Diagram** objects will work with **DiagramPage** objects too. A DiagramDocument forwards each event raised by its pages, giving you a chance to use the same handler for events raised by different pages. If a page should implement distinct user interaction or validation logic, it is also possible to attach handlers to the **Diagram** events inherited by **DiagramPage**.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramBase
  MindFusion.Diagramming.Diagram
    MindFusion.Diagramming.DiagramPage

See Also

DiagramPage Members
MindFusion.Diagramming Namespace

DiagramPage Members

Represents a single page in a DiagramDocument object.

The following tables list the members exposed by the DiagramPage type.

Public Constructors

	Name	Description
	DiagramPage	Initializes a new instance of the DiagramPage class.

Public Fields

	Name	Description
	Dummy	A shared DummyNode instance. (Inherited from Diagram.)

Public Properties

	Name	Description
	ActiveItem	Gets or sets the active diagram element. (Inherited from Diagram.)
	ActiveItemHandlesStyle	Gets the visual style of adjustment handles of the active item. (Inherited from Diagram.)
	ActiveLayer	Gets or sets the zero-based index of the active layer. (Inherited from Diagram.)
	AdjustmentHandlesSize	Gets or sets the size of the handles, which appear around selected objects. (Inherited from Diagram.)
	AlignmentGuidePen	Gets or sets the pen used to draw alignment guides. (Inherited from Diagram.)
	AlignToGrid	Gets or sets a value indicating whether the diagram items is aligned to a grid. (Inherited from Diagram.)
	AllowLinksRepeat	Gets or sets a value indicating whether more than one link can connect the same origin and destination diagram nodes. (Inherited from Diagram.)
	AllowMultipleResize	Gets or sets whether multiple selected nodes can be resized simultaneously. (Inherited from Diagram.)
	AllowSelfLoops	Gets or sets a value indicating whether self-loops are allowed. (Inherited from Diagram.)
	AllowSplitLinks	Gets or sets a value indicating whether link segments can be added and removed interactively. (Inherited from Diagram.)
	AllowUnanchoredLinks	Gets or sets a value indicating whether links can be connected to nodes to which there aren't anchor points assigned. (Inherited from Diagram.)
	AllowUnconnectedLinks	Gets or sets a value indicating whether users are allowed to draw links that are not connected to any node. (Inherited from Diagram.)
	AutoAlignDistance	Gets or sets the maximal horizontal or vertical distance at which a node aligns to the borders of other nodes. (Inherited from Diagram.)
	AutoAlignNodes	Gets or sets whether nodes should automatically align to the borders of other nodes during user interaction. (Inherited from Diagram.)

	AutoArrangeAvoidContainers	Gets or sets a value indicating whether auto-arranged link labels should avoid containers. (Inherited from Diagram.)
	AutoArrangeAvoidSegments	Gets or sets a value indicating whether auto-arranged link labels should avoid link segments. (Inherited from Diagram.)
	AutoCloseDistance	Gets or sets the maximum distance between first and last points of a FreeFormNode for which the node's outline is closed automatically. (Inherited from Diagram.)
	AutoDeleteLinks	Gets or sets whether links are deleted automatically when their Origin or Destination is deleted. (Inherited from Diagram.)
	AutoHighlightRows	Gets or sets whether a table row should be highlighted when the user clicks on it. (Inherited from Diagram.)
	AutoResize	Gets or sets a value indicating whether the document area must be resized automatically so it fits the diagram contents. (Inherited from Diagram.)
	AutoSnapDistance	Specifies the maximum distance at which links auto-snap to nodes. (Inherited from Diagram.)
	AutoSnapLinks	Gets or sets the default value for the AutoSnapToNode property of new links. (Inherited from Diagram.)
	BackBrush	Gets or sets the brush used for painting the diagram background. (Inherited from Diagram.)
	BackgroundImage	Gets or sets the image displayed as background of this diagram. (Inherited from Diagram.)
	BackgroundImageAlign	Gets or sets a value indicating how the background image is positioned, relatively to the scrollable document area. (Inherited from Diagram.)
	BackgroundImageUrl	URL of the image painted as diagram's background. (Inherited from Diagram.)
	BarcodeNodeStyle	Gets the theme-level style for barcode nodes. (Inherited from Diagram.)
	Bounds	Gets or sets a value representing the size of the diagram's scrollable area. (Inherited from Diagram.)
	BoundsPen	The pen with which the document bounds should be drawn. (Inherited from Diagram.)
	BranchIndicatorColor	Gets or sets the color of link branch indicators. (Inherited from Diagram.)
	BranchIndicatorSize	Gets or sets the size of link branch indicators. (Inherited from Diagram.)
	CellCustomDraw	Gets or sets the default type of custom painting done on the cells of tables. (Inherited from Diagram.)
	CellFrameStyle	Gets or sets the default cell border style of new tables. (Inherited from Diagram.)
	CompositeNodeStyle	Gets the theme-level style for composite nodes. (Inherited from Diagram.)
	ContainerCaption	Gets or sets the default value for the Caption property of containers. (Inherited from Diagram.)
	ContainerCaptionHeight	Gets or sets the default value for the CaptionHeight property of containers. (Inherited from Diagram.)
	ContainerHighlightPen	Gets or sets the default value for the HighlightPen property of containers. (Inherited from Diagram.)
	ContainerMargin	Gets or sets the default value for the Margin property of containers. (Inherited from Diagram.)
	ContainerMinimumSize	Gets or sets the default value for the MinimumSize property of containers. (Inherited from Diagram.)

	ContainerNodeStyle	Gets the theme-level style for container nodes. (Inherited from Diagram.)
	ContainersFoldable	Gets or sets the default value for the Foldable property of containers. (Inherited from Diagram.)
	CrossingRadius	Gets or sets the radius length of decorations displayed at link intersection points. (Inherited from Diagram.)
	DefaultShape	Gets or sets the default shape of shape nodes. (Inherited from Diagram.)
	DiagramLinkStyle	Gets the theme-level style for diagram links. (Inherited from Diagram.)
	DiagramStyle	Gets the theme-level style for the diagram. (Inherited from Diagram.)
	Dirty	Gets or sets a value indicating whether the diagram has changed since the last save. (Inherited from Diagram.)
	DisabledHandlesStyle	Gets the visual style of disabled adjustment handles. (Inherited from Diagram.)
	DynamicLinks	Gets or sets the default value for the Dynamic property of new links. (Inherited from Diagram.)
	EffectiveBackBrush	Gets the currently effective background brush of the diagram. (Inherited from Diagram.)
	EffectsProvider	Gets or sets an object used to generate visual effects for shape nodes such as embossment and shadow dispersion. (Inherited from Diagram.)
	EnableEmbossEffects	Gets or sets a value indicating whether emboss effect for shape nodes is enabled. (Inherited from Diagram.)
	EnableLanes	Gets or sets whether lanes are enabled. (Inherited from Diagram.)
	EnableShadowEffects	Gets or sets a value indicating whether the shadow dispersion effect for nodes is enabled. (Inherited from Diagram.)
	EnableSpatialIndex	Gets or sets whether to index item positions for faster hit-testing and viewport clipping queries. (Inherited from Diagram.)
	EnableStyledText	Gets or sets the default value for the EnableStyledText property of new nodes. (Inherited from Diagram.)
	ExpandButtonAction	Gets or sets a value indicating what should happen when a user clicks the [±] button that is displayed near expandable nodes. (Inherited from Diagram.)
	ExpandButtonPosition	Gets or sets where expand/collapse buttons are located relatively to their nodes. (Inherited from Diagram.)
	ExpandOnIncoming	Gets or sets a value, indicating in which direction object hierarchies are expanded. (Inherited from Diagram.)
	ExteriorBrush	Gets or sets the brush used for painting the area outside **Bounds**. (Inherited from Diagram.)
	Factory	Gets a Factory instance that lets you add programmatically new items to the diagram. (Inherited from Diagram.)
	FocusedComponent	Gets or sets the component within this diagram which currently has the input focus. (Inherited from Diagram.)
	FoldedContainerSize	Gets or sets the size of folded containers. (Inherited from Diagram.)
	Font	Gets or sets the default font for new diagram items. (Inherited from Diagram.)
	FreeFormAttractDistance	Gets or sets the radius around dragged free-form adjustment handle in which other points of FreeFormNode are modified too. (Inherited from Diagram.)
	FreeFormTargets	An array of Shape objects used to replace FreeFormNode instances with ShapeNode ones when Behavior is set to LinkFreeShapes or DrawFreeShapes. (Inherited from Diagram.)

	GridColor	Gets or sets the color of the grid points. (Inherited from Diagram.)
	GridOffsetX	Gets or sets the horizontal offset of the first point of the alignment grid. (Inherited from Diagram.)
	GridOffsetY	Gets or sets the vertical offset of the first point of the alignment grid. (Inherited from Diagram.)
	GridPatternHatch	Gets or sets a hatch style used to draw dense alignment grid at low zoom levels. (Inherited from Diagram.)
	GridPatternThreshold	Gets or sets the distance between grid points (in pixels) that triggers drawing using **GridPatternHatch**. (Inherited from Diagram.)
	GridSizeX	Gets or sets the horizontal distance between adjacent grid points. (Inherited from Diagram.)
	GridSizeY	Gets or sets the vertical distance between adjacent grid points. (Inherited from Diagram.)
	GridStyle	Gets or sets the visual style of the alignment grid. (Inherited from Diagram.)
	Groups	Gets a collection of all groups in the diagram. (Inherited from Diagram.)
	HitTestPriority	Gets or sets a value indicating the order in which items are hit-tested. (Inherited from Diagram.)
	iconUrl	Gets or sets URL of the image to paint as the page's icon.
	Interaction	Gets an InteractionState instance representing the current action being carried out by the user. (Inherited from Diagram.)
	IsProEdition	Gets a value indicating whether the currently loaded MvcDiagram is a professional version of the control. (Inherited from Diagram.)
	Items	Gets a collection of all items in this diagram. (Inherited from Diagram.)
	LaneGrid	Gets the Grid instance that represents the lane grid. (Inherited from Diagram.)
	Layers	Gets a list of the layers in the current diagram. (Inherited from Diagram.)
	LinkBaseShape	Gets or sets the shape displayed at the base of new links. (Inherited from Diagram.)
	LinkBaseShapeSize	Gets or sets the default size of the base of new links. (Inherited from Diagram.)
	LinkBranchIndicator	Gets or sets a value indicating what shapes to display as markers at points where overlapping Cascading links diverge. (Inherited from Diagram.)
	LinkBrush	Gets or sets the default value for the Brush property of links. (Inherited from Diagram.)
	LinkCascadeOrientation	Gets or sets the initial orientation of the first segment of new cascading links. (Inherited from Diagram.)
	LinkCrossings	Gets or sets how to render the intersection points where links cross their paths. (Inherited from Diagram.)
	LinkCustomDraw	Gets or sets the default type of custom painting done on links. (Inherited from Diagram.)
	LinkEndsMovable	Gets or sets a value indicating whether link ends can be moved by users after the link is created. (Inherited from Diagram.)
	LinkHandlesStyle	Gets or sets the default value for the HandlesStyle property of new links. (Inherited from Diagram.)
	LinkHeadShape	Gets or sets the shape displayed as head of new links. (Inherited from Diagram.)

	LinkHeadShapeSize	Gets or sets the default size of arrowheads. (Inherited from Diagram.)
	LinkHitDistance	Gets or sets a value specifying how far from a link a click is still considered a hit. (Inherited from Diagram.)
	LinkIntermediateShape	Gets or sets the default shape to be displayed in the middle of each segment of links. (Inherited from Diagram.)
	LinkIntermediateShapeSize	Gets or sets the default size of the shapes at middles of link segments. (Inherited from Diagram.)
	LinkMergeIndicator	Gets or sets a value indicating what shapes to display as markers at points where overlapping **Cascading** links merge. (Inherited from Diagram.)
	LinkPen	Gets or sets the default value for the Pen and HeadPen properties of links. (Inherited from Diagram.)
	LinkRouter	Gets or sets the component used to find paths for auto-routed links when diagram nodes are added or their positions change. (Inherited from Diagram.)
	Links	Gets a collection of all DiagramLink objects in the diagram. (Inherited from Diagram.)
	LinkSegments	Gets or sets the default number of link segments. (Inherited from Diagram.)
	LinkShape	Gets or sets the default value of the Shape property of new links. (Inherited from Diagram.)
	LinksRetainForm	Gets or sets whether links should maintain their forms if the nodes linked with those links are moved around. (Inherited from Diagram.)
	LinkText	Gets or sets the default text of links. (Inherited from Diagram.)
	LinkTextStyle	Gets or sets default placement and orientation for links' text. (Inherited from Diagram.)
	MeasureTextRenderingHint	Gets or sets the TextRenderingHint to use when measuring texts. This should be effectively set to the value of DiagramView.TextRenderingHint. (Inherited from Diagram.)
	MeasureUnit	Gets or sets the unit of measure used for logical coordinates in this diagram. (Inherited from Diagram.)
	MergeIndicatorColor	Gets or sets the color of link merge indicators. (Inherited from Diagram.)
	MergeIndicatorSize	Gets or sets the size of link merge indicators. (Inherited from Diagram.)
	MergeThreshold	Gets or sets the maximum distance between adjacent control points of a link at which the respective segments can be merged. (Inherited from Diagram.)
	MinimumNodeSize	Gets or sets the minimum node size that the user is allowed to set. (Inherited from Diagram.)
	ModificationEffect	Gets or sets the visual effect to apply during interactive modification of items. (Inherited from Diagram.)
	NodeEffects	Gets a collection with all effects applied to the nodes of this diagram. (Inherited from Diagram.)
	Nodes	Gets a collection of all DiagramNode objects in the diagram. (Inherited from Diagram.)
	NodesExpandable	Gets or sets a value indicating whether newly created nodes can be expanded. (Inherited from Diagram.)
	NowLoading	Gets a value indicating whether the diagram data is now being loaded from a file. (Inherited from Diagram.)
	PolygonalTextLayout	Gets or sets a value indicating whether to enable the formatting of text inside non-rectangular shape node outlines. (Inherited from Diagram.)

	RecursiveExpand	Gets or sets value indicating how a tree is expanded. (Inherited from Diagram.)
	RenderOptions	Gets the current rendering options for this diagram. (Inherited from Diagram.)
	RestrictItemsToBounds	Gets or sets a value indicating whether users are allowed to place items outside the scrollable area of the diagram. (Inherited from Diagram.)
	RoundedLinks	Gets or sets a value indicating whether the joins between segments of **Polyline** and **Cascading** links should be rounded. (Inherited from Diagram.)
	RoundedLinksRadius	Gets or sets the radius of arc joins of rounded links' segments. (Inherited from Diagram.)
	RoundRectFactor	Gets or sets the relative size of arcs displayed at the corners of rounded rectangles. (Inherited from Diagram.)
	RouteLinks	Gets or sets a value indicating whether links should avoid nodes by going the shortest path from their origin to their destination without crossing any other nodes. (Inherited from Diagram.)
	RoutingOptions	Gets an object, which allows customizing various parameters of the link routing algorithm. (Inherited from Diagram.)
	RowHighlightBrush	Gets or sets the Brush used to fill highlighted table rows. (Inherited from Diagram.)
	ScriptHelper	Gets a ScriptHelper object which can be used in html scripts to instantiate various objects and to call static methods. (Inherited from Diagram.)
	SelectAfterCreate	Gets or sets a value indicating whether newly created objects are automatically selected. (Inherited from Diagram.)
	SelectedItemHandlesStyle	Gets the visual style of adjustment handles of selected items. (Inherited from Diagram.)
	Selection	Gets a Selection instance that represents the selection of items in this diagram. (Inherited from Diagram.)
	SelectionOnTop	Gets or sets a value indicating whether selected items should be painted on top of other items. (Inherited from Diagram.)
	SerializationBinder	Gets or sets a SerializationBinder object used to control class loading. (Inherited from Diagram.)
	ShadowBrush	Gets or sets the default brush used to paint the shadows cast by diagram items. (Inherited from Diagram.)
	ShadowOffsetX	Gets or sets the horizontal offset of item shadows. (Inherited from Diagram.)
	ShadowOffsetY	Gets or sets the vertical offset of item shadows. (Inherited from Diagram.)
	ShadowsStyle	Gets or sets the style for rendering shadows. (Inherited from Diagram.)
	ShapeBrush	Gets or sets the default value for the **Brush** property of shape nodes. (Inherited from Diagram.)
	ShapeCustomDraw	Gets or sets the default type of custom painting done for shape nodes. (Inherited from Diagram.)
	ShapeHandlesStyle	Gets or sets the default value for the **HandlesStyle** property of new shape nodes. (Inherited from Diagram.)
	ShapeNodeStyle	Gets the theme-level style for shape nodes. (Inherited from Diagram.)
	ShapeOrientation	Gets or sets the default value for the ShapeOrientation property of shape nodes. (Inherited from Diagram.)
	ShapePen	Gets or sets the default value for the **Pen** property of shape nodes. (Inherited from Diagram.)

	ShapeText	Gets or sets the default text of shape nodes. (Inherited from Diagram.)
	ShowAnchors	Gets or sets a value indicating whether and when node anchor points should be displayed. (Inherited from Diagram.)
	ShowDisabledHandles	Gets or sets a value indicating whether manipulation handles are painted when they are disabled. (Inherited from Diagram.)
	ShowGrid	Gets or sets a value indicating whether the alignment grid is visible. (Inherited from Diagram.)
	ShowHandlesOnDrag	Gets or sets a value indicating whether to display selection handles of objects under the mouse while another object is being modified. (Inherited from Diagram.)
	SnapToAnchor	Gets or sets when link end points should be aligned to the nearest anchor points. (Inherited from Diagram.)
	SortGroupsByZ	Gets or sets a value indicating whether grouped items should be drawn sorted by their z-order while the whole group is being moved. (Inherited from Diagram.)
	Style	Gets or sets the style associated with the diagram. (Inherited from Diagram.)
	SvgNodeStyle	Gets the theme-level style for SVG nodes. (Inherited from Diagram.)
	TableBrush	Gets or sets the default value for the **Brush** property of tables. (Inherited from Diagram.)
	TableCaption	Gets or sets the default text for tables' captions. (Inherited from Diagram.)
	TableCaptionHeight	Gets or sets the default height of table captions. (Inherited from Diagram.)
	TableColumnCount	Gets or sets the default number of table columns. (Inherited from Diagram.)
	TableColumnWidth	Gets or sets the default width of tables' columns. (Inherited from Diagram.)
	TableConnectionStyle	Gets or sets a value indicating whether drawing a link between tables should connect the rows of those tables, or the tables as whole entities. (Inherited from Diagram.)
	TableCustomDraw	Gets or sets the default type of custom painting done on tables. (Inherited from Diagram.)
	TableHandlesStyle	Gets or sets the default value for the **HandlesStyle** property of new tables. (Inherited from Diagram.)
	TableNodeStyle	Gets the theme-level style for table nodes. (Inherited from Diagram.)
	TablePen	Gets or sets the default value for the **Pen** property of tables. (Inherited from Diagram.)
	TableRowCount	Gets or sets the default number of table rows. (Inherited from Diagram.)
	TableRowHeight	Gets or sets the default height of tables' rows. (Inherited from Diagram.)
	TableShape	Gets or sets the default value of the Shape property of new tables. (Inherited from Diagram.)
	TablesScrollable	Gets or sets the default value for the Scrollable property of new tables. (Inherited from Diagram.)
	Tag	Gets or sets custom data associated with this diagram. (Inherited from Diagram.)
	TextColor	Gets or sets the default color of text displayed inside diagram items. (Inherited from Diagram.)

	TextFormat	Gets or sets the default alignment and formatting style for text displayed inside nodes and table cells. (Inherited from Diagram.)
	Theme	Gets or sets the theme associated with the diagram. (Inherited from Diagram.)
	title	Gets or sets the page title.
	titleColor	Gets or sets the color of the title when the title is displayed.
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram. (Inherited from Diagram.)
	TouchHitDistance	Specifies distance from adjustment handles at which they are still considered as hit when processing touch input. (Inherited from Diagram.)
	TouchThreshold	Gets or sets the threshold value for easier touch interaction through adjustment handles. (Inherited from Diagram.)
	TreeViewNodeStyle	Gets the theme-level style for tree view nodes. (Inherited from Diagram.)
	UndoManager	Gets a reference to the diagram's UndoManager instance. (Inherited from Diagram.)
	ValidityChecks	Gets or sets a value indicating whether validity checks are performed each time an item is added to the diagram. (Inherited from Diagram.)
	VisioNodeStyle	Gets the theme-level style for Visio nodes. (Inherited from Diagram.)
	XmlOptions	An XmlOptions object providing properties for customizing the XML serialization process. (Inherited from DiagramBase.)

Public Methods

	Name	Description
	AlignPointToGrid	Returns a point of the alignment grid nearest to the one passed as an argument. (Inherited from Diagram.)
	AllowNodeBounds	Determines whether node can be placed at specified coordinates. (Inherited from Diagram.)
	Arrange	Arranges the diagram using the specified graph layout algorithm. (Inherited from Diagram.)
	ArrangeAnimated	Arranges the diagram using the specified graph layout algorithm, and animates items moving to their new positions. (Inherited from Diagram.)
	ArrangeAsync	Arranges the diagram asynchronously using the specified graph layout algorithm. (Inherited from Diagram.)
	ArrangeLinkLabels	Rearranges link labels whose AutoArrange property is enabled. (Inherited from Diagram.)
	ChangeUnit	Sets new **MeasureUnit** value and scales the coordinates of diagram items so that their absolute size remains unchanged. (Inherited from Diagram.)
	ClearAll	Removes all items from the diagram. (Inherited from Diagram.)
	Clone	Creates a copy of this diagram. (Inherited from Diagram.)
	CopySelection	Overloaded. Creates a copy of specified diagram elements. (Inherited from Diagram.)
	CopyTo	Copies this diagram's properties and items to specified target diagram. (Inherited from Diagram.)
	CreateImage	Overloaded. Creates a bitmap containing an image of the diagram. (Inherited from Diagram.)
	CreateMeasureGraphics	Creates an IGraphics object that can be used to obtain various string size measurements. (Inherited from Diagram.)

	CreateThemeFromDefaults	Creates a Theme object representing the default properties of the diagram. (Inherited from Diagram.)
	Draw	Renders the diagram on the specified **IGraphics** object. (Inherited from Diagram.)
	DrawAlignmentGuides	Draws the current alignment guides. (Inherited from Diagram.)
	DrawInteraction	Draws the items that the user is currently modifying. (Inherited from Diagram.)
	DrawInteractions	For internal use. (Inherited from Diagram.)
	DrawLayer	Renders all diagram items to the specified graphics. (Inherited from Diagram.)
S	DrawStyledText	Draws a formatted text string, the same way text is drawn in shape nodes whose **EnableStyledText** property is set to **true**. This method lets you display styled text in custom-drawn nodes or links. (Inherited from Diagram.)
	EffectiveTouchHitDistance	Returns **TouchHitDistance** or a default value. (Inherited from Diagram.)
	EnumByReverseZ	Hit-tests diagram items in reverse Z order. (Inherited from Diagram.)
	EnumTopLevelByReverseZ	Hit-tests diagram top-level items in reverse Z order. (Inherited from Diagram.)
	ExecuteCommand	Executes the specified command on this diagram. (Inherited from Diagram.)
	FindGroup	Finds the group that has the specified tag value. (Inherited from Diagram.)
	FindLink	Finds the link that has the specified tag value. (Inherited from Diagram.)
	FindLinkById	Finds the link that has the specified id value. (Inherited from Diagram.)
	FindNode	Finds the node that has the specified tag value. (Inherited from Diagram.)
	FindNodeById	Finds the node that has the specified id value. (Inherited from Diagram.)
	GetContentBounds	Returns the smallest rectangle that bounds all diagram objects. (Inherited from Diagram.)
	GetItemAt	Overloaded. (Inherited from Diagram.)
	GetLinkAt	Overloaded. Finds and returns the link that lies at the specified location. (Inherited from Diagram.)
	GetLinksAt	Returns the links that contain the specified point. (Inherited from Diagram.)
	GetNearestNode	Returns the node nearest to the specified point. (Inherited from Diagram.)
	GetNodeAt	Overloaded. Gets the node that lies at the specified location. (Inherited from Diagram.)
	GetNodesAt	Returns the nodes that contain the specified point. (Inherited from Diagram.)
	GetNodesInViewport	Returns all nodes visible in the current viewport. (Inherited from Diagram.)
	HandleKeyDown	Propagates keyboard input to interested manipulators and components. (Inherited from Diagram.)
	HandleKeyPress	Propagates keyboard input to interested manipulators. (Inherited from Diagram.)
	HandleKeyUp	Propagates keyboard input to interested components. (Inherited from Diagram.)

	HandleMouseMove	Propagates mouse input to interested manipulators. (Inherited from Diagram.)
	HitTestComponents	Tests whether the specified point is within a component of a diagram item and returns this component. (Inherited from Diagram.)
	HitTestLabels	Returns the item label located at specified position. (Inherited from Diagram.)
	HitTestManipulators	Tests whether the specified point is within the manipulator of a diagram object and triggers the manipulator. (Inherited from Diagram.)
	Invalidate	Overloaded. Invalidates the diagram or a region of the diagram, causing it to be repainted. (Inherited from Diagram.)
	InvalidateZOrder	Forces a new sort of the Z order list before next redraw. (Inherited from Diagram.)
S	IsClassRegistered	Checks if a class is registered for serialization support. (Inherited from Diagram.)
S	IsClientClassRegistered	Checks if a class is registered for JSON serialization support. (Inherited from Diagram.)
	IsItemInteractive	Checks if the specified item can be modified interactively. (Inherited from Diagram.)
	IsItemLocked	Checks if the specified item is locked. (Inherited from Diagram.)
	IsItemVisible	Checks if the specified item is currently visible. (Inherited from Diagram.)
	LoadFromJson	Loads the diagram from a JSON-formatted string. (Inherited from Diagram.)
	LoadFromJsonAsync	Loads the diagram from a JSON-formatted string. (Inherited from Diagram.)
	LoadFromJsonFile	Loads the diagram from a JSON file. (Inherited from Diagram.)
	LoadFromJsonFileAsync	Loads the diagram from a JSON file. (Inherited from Diagram.)
	LoadFromJsonObject	Loads the diagram from a JSON object. (Inherited from Diagram.)
	LoadFromString	Decodes diagram contents from a string that was created with SaveToString. (Inherited from Diagram.)
	LoadFromXml	Overloaded. Loads the contents of an XML diagram document into the control. (Inherited from Diagram.)
	LoadFromXmlAsync	Overloaded. (Inherited from Diagram.)
	LoadSelectionFromJson	(Inherited from Diagram.)
	MeasureString	Overloaded. Measures a text taking into consideration the current affine transform of the diagram. (Inherited from Diagram.)
	Navigate	Selects the nearest item in the specified direction. (Inherited from Diagram.)
	NewInstance	Custom Diagram classes must override this method to return an instance of their type. (Inherited from Diagram.)
	PasteSelection	Adds the specified items and groups to the specified diagram document. (Inherited from Diagram.)
	RaiseCellTextEdited	Raises the CellTextEdited event . (Inherited from Diagram.)
	RaiseCellTextEditing	Raises the CellTextEditing event to validate entering in-place editing mode. (Inherited from Diagram.)
	RaiseClicked	Overloaded. Raises the Clicked event of the diagram or one of the *Clicked events related to diagram items. (Inherited from Diagram.)

	RaiseCreated	Raises the appropriate *Created event according to the type of the specified item. (Inherited from Diagram.)
	RaiseDeleting	Raises the appropriate *Deleting event according to the type of the specified item. (Inherited from Diagram.)
	RaiseDoubleClicked	Overloaded. Raises the DoubleClicked event of the diagram or one of the *DoubleClicked events related to diagram items. (Inherited from Diagram.)
	RaiseLinkTextEdited	Raises the LinkTextEdited event. (Inherited from Diagram.)
	RaiseLinkTextEditing	Raises the LinkTextEditing event to validate entering inplace edit mode. (Inherited from Diagram.)
	RaiseModified	Raises the NodeModified event. (Inherited from Diagram.)
	RaiseMouseHover	Raises the appropriate Pointed event. (Inherited from Diagram.)
	RaiseNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	RaiseNodeTextEdited	Raises the NodeTextEdited event. (Inherited from Diagram.)
	RaiseNodeTextEditing	Raises the NodeTextEditing event to validate entering inplace edit mode. (Inherited from Diagram.)
	RaisePasted	Raises the appropriate Pasted event. (Inherited from Diagram.)
	RaiseSelecting	Raises the appropriate Selecting event. (Inherited from Diagram.)
	RaiseTreeItemTextEditing	Raises the TreeItemTextEditing event to validate entering inplace edit mode. (Inherited from Diagram.)
	RaiseViewRemoved	Raises the ViewRemoved event. (Inherited from Diagram.)
	RegisterClass	Overloaded. (Inherited from Diagram.)
	RegisterDefaultStyle	Registers a default style for the objects of the specified type. (Inherited from Diagram.)
	RegisterItemClass	Overloaded. (Inherited from Diagram.)
	ResetAlignmentGuides	For internal use. (Inherited from Diagram.)
	ResizeToFitItem	Resizes the document extents so that the specified item is contained within. (Inherited from Diagram.)
	ResizeToFitItems	Overloaded. Resizes the document scrollable area so it fits all diagram items. (Inherited from Diagram.)
	ResumeLinkRouter	Called when the control ends a batch update operation that might affect many links. (Inherited from Diagram.)
	RouteAllLinks	Overloaded. (Inherited from Diagram.)
	SaveSelectionToJson	Saves the diagram selection to a JSON object. (Inherited from Diagram.)
	SaveToJson	Saves the diagram to a JSON-formatted string. (Inherited from Diagram.)
	SaveToJsonAsync	Saves the diagram to a JSON-formatted string. (Inherited from Diagram.)
	SaveToJsonFile	Saves the diagram into a JSON file. (Inherited from Diagram.)
	SaveToJsonFileAsync	Saves the diagram into a JSON file. (Inherited from Diagram.)
	SaveToJsonObject	Saves the diagram to a JSON object. (Inherited from Diagram.)
	SaveToString	Encodes the diagram contents into an ASCII string using the specified format and optionally clearing the **Dirty** flag. (Inherited from Diagram.)
	SaveToXml	Overloaded. Saves the diagram to an XML document. (Inherited from Diagram.)
	SaveToXmlAsync	Overloaded. (Inherited from Diagram.)

	SetShadowOffset	Sets the default offset at which items shadows are drawn. (Inherited from Diagram.)
	SuspendLinkRouter	Called when the control starts a batch update operation that might affect many links. The diagram collects links specified through Route and RouteAllLinks calls and defers finding their paths to when **ResumeLinkRouter** is called. (Inherited from Diagram.)
	TrackMouseHover	Specifies whether the diagram view should track mouse hover events. (Inherited from DiagramBase.)
	UpdateMeasureTextScale	Gets or sets the scale to use when measuring texts. This should be effectively set to the value of DiagramView.Zoom / 100. (Inherited from Diagram.)
	UpdateRuntimeIndices	For internal use. (Inherited from Diagram.)
	UpdateZOrder	Immediately sorts the Z-order list if Z-order has been invalidated. Otherwise it will be sorted next time the diagram draws its contents. (Inherited from Diagram.)

Protected Methods

	Name	Description
	AddItem	Invoked by the AddItemCmd and RemoveItemCmd classes to effectively add the specified item to the diagram. (Inherited from Diagram.)
	CreateAddItemCmd	Creates an **AddItemCmd** instance that implements undo/redo of adding an item to the diagram. (Inherited from Diagram.)
	CreateModifyItemCmd	Creates a ModifyItemCmd instance that implements undo/redo of item move and resize operations. (Inherited from Diagram.)
	CreateRemoveItemCmd	Creates a **RemoveItemCmd** instance that implements undo/redo of removing an item from the diagram. (Inherited from Diagram.)
	DeleteGroup	Removes the specified group from the group collection. (Inherited from Diagram.)
	Dispose	Clean up any resources being used. (Inherited from Diagram.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from Diagram.)
	OnActionRecorded	Raises the ActionRecorded event. (Inherited from DiagramBase.)
	OnActionRecording	Raises the ActionRecording event. (Inherited from DiagramBase.)
	OnActionRedone	Raises the ActionRedone event. (Inherited from DiagramBase.)
	OnActionUndone	Raises the ActionUndone event. (Inherited from DiagramBase.)
	OnBeginLoad	Raises the BeginLoad event. (Inherited from DiagramBase.)
	OnBoundsChanged	Raises the BoundsChanged event. (Inherited from DiagramBase.)
	OnCellClicked	Raises the CellClicked event. (Inherited from DiagramBase.)
	OnCellDoubleClicked	Raises the CellDoubleClicked event. (Inherited from DiagramBase.)
	OnCellTextEdited	Raises the **CellTextEdited** event. (Inherited from DiagramBase.)
	OnCellTextEditing	Raises the **CellTextEditing** event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnClicked	Raises the **Clicked** event. (Inherited from DiagramBase.)
	OnContainerChildAdded	Raises the ContainerChildAdded event. (Inherited from DiagramBase.)
	OnContainerChildRemoved	Raises the ContainerChildRemoved event. (Inherited from DiagramBase.)
	OnContainerFolded	Raises the ContainerFolded event. (Inherited from DiagramBase.)

	OnContainerPageChanged	Raises the ContainerPageChanged event. (Inherited from DiagramBase.)
	OnContainerUnfolded	Raises the ContainerUnfolded event. (Inherited from DiagramBase.)
	OnContentVisibilityChanged	Raises the ContentVisibilityChanged event. (Inherited from DiagramBase.)
	OnDefaultShapeChanged	Raises the DefaultShapeChanged event. (Inherited from DiagramBase.)
	OnDeserializeTag	Raises the DeserializeTag event. (Inherited from DiagramBase.)
	OnDestinationAnchorChanged	Raises the DestinationAnchorChanged event. (Inherited from DiagramBase.)
	OnDirtyChanged	Raises the DirtyChanged event. (Inherited from DiagramBase.)
	OnDoubleClicked	Raises the **DoubleClicked** event. (Inherited from DiagramBase.)
	OnDrawAdjustmentHandles	Raises the DrawAdjustmentHandles event. (Inherited from DiagramBase.)
	OnDrawAnchorPoint	Raises the DrawAnchorPoint event. (Inherited from DiagramBase.)
	OnDrawBackground	Raises the DrawBackground event. (Inherited from DiagramBase.)
	OnDrawBranchIndicator	Raises the DrawBranchIndicator event. (Inherited from DiagramBase.)
	OnDrawCell	Raises the DrawCell event. (Inherited from DiagramBase.)
	OnDrawForeground	Raises the DrawForeground event. (Inherited from DiagramBase.)
	OnDrawLink	Raises the DrawLink event. (Inherited from DiagramBase.)
	OnDrawNode	Raises the DrawNode event. (Inherited from DiagramBase.)
	OnEndLoad	Raises the EndLoad event. (Inherited from DiagramBase.)
	OnExpandButtonClicked	Raises the ExpandButtonClicked event. (Inherited from DiagramBase.)
	OnGroupDestroyed	Raises the GroupDestroyed event. (Inherited from DiagramBase.)
	OnHitTestAdjustmentHandles	Raises the HitTestAdjustmentHandles event. (Inherited from DiagramBase.)
	OnInitializeLasso	Raises the InitializeLasso event. (Inherited from DiagramBase.)
	OnInitializeLink	Raises the InitializeLink event. (Inherited from DiagramBase.)
	OnInitializeNode	Raises the InitializeNode event. (Inherited from DiagramBase.)
	OnItemAdded	Raises the ItemAdded event. (Inherited from DiagramBase.)
	OnItemRemoved	Raises the ItemRemoved event. (Inherited from DiagramBase.)
	OnJsonDeserializeTag	Raises the JsonDeserializeTag event. (Inherited from DiagramBase.)
	OnJsonSerializeTag	Raises the JsonSerializeTag event. (Inherited from DiagramBase.)
	OnLayerVisibilityChanged	Raises the LayerVisibilityChanged event. (Inherited from DiagramBase.)
	OnLinkActivated	Raises the LinkActivated event. (Inherited from DiagramBase.)
	OnLinkClicked	Raises the LinkClicked event. (Inherited from DiagramBase.)
	OnLinkCreateCancelled	Raises the LinkCreateCancelled event. (Inherited from DiagramBase.)
	OnLinkCreated	Raises the LinkCreated event. (Inherited from DiagramBase.)
	OnLinkCreating	Raises the LinkCreating event. (Inherited from DiagramBase.)
	OnLinkDeactivated	Raises the LinkDeactivated event. (Inherited from DiagramBase.)
	OnLinkDeleted	Raises the LinkDeleted event. (Inherited from DiagramBase.)
	OnLinkDeleting	Raises the LinkDeleting event. (Inherited from DiagramBase.)

	OnLinkDeselected	Raises the LinkDeselected event. (Inherited from DiagramBase.)
	OnLinkDoubleClicked	Raises the LinkDoubleClicked event. (Inherited from DiagramBase.)
	OnLinkHovered	Raises the LinkHovered event. (Inherited from DiagramBase.)
	OnLinkModified	Raises the LinkModified event. (Inherited from DiagramBase.)
	OnLinkModifyCancelled	Raises the LinkModifyCancelled event. (Inherited from DiagramBase.)
	OnLinkModifying	Raises the LinkModifying event. (Inherited from DiagramBase.)
	OnLinkPasted	Raises the LinkPasted event. (Inherited from DiagramBase.)
	OnLinkRouted	Raises the LinkRouted event. (Inherited from DiagramBase.)
	OnLinkSelected	Raises the LinkSelected event. (Inherited from DiagramBase.)
	OnLinkSelecting	Raises the LinkSelecting event. (Inherited from DiagramBase.)
	OnLinkSplit	Raises the LinkSplit event. (Inherited from DiagramBase.)
	OnLinkStartModifying	Raises the LinkStartModifying event. (Inherited from DiagramBase.)
	OnLinkTextEdited	Raises the **LinkTextEdited** event. (Inherited from DiagramBase.)
	OnLinkTextEditing	Raises the **LinkTextEditing** event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnMeasureUnitChanged	Raises the MeasureUnitChanged event. (Inherited from DiagramBase.)
	OnNodeActivated	Raises the NodeActivated event. (Inherited from DiagramBase.)
	OnNodeClicked	Raises the NodeClicked event. (Inherited from DiagramBase.)
	OnNodeCreateCancelled	Raises the NodeCreateCancelled event. (Inherited from DiagramBase.)
	OnNodeCreated	Raises the NodeCreated event. (Inherited from DiagramBase.)
	OnNodeCreating	Raises the NodeCreating event. (Inherited from DiagramBase.)
	OnNodeDeactivated	Raises the NodeDeactivated event. (Inherited from DiagramBase.)
	OnNodeDeleted	Raises the NodeDeleted event. (Inherited from DiagramBase.)
	OnNodeDeleting	Raises the NodeDeleting event. (Inherited from DiagramBase.)
	OnNodeDeselected	Raises the NodeDeselected event. (Inherited from DiagramBase.)
	OnNodeDoubleClicked	Raises the NodeDoubleClicked event. (Inherited from DiagramBase.)
	OnNodeHovered	Raises the NodeHovered event. (Inherited from DiagramBase.)
	OnNodeModified	Raises the **NodeModified** event. (Inherited from DiagramBase.)
	OnNodeModifyCancelled	Raises the NodeModifyCancelled event. (Inherited from DiagramBase.)
	OnNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	OnNodePasted	Raises the NodePasted event. (Inherited from DiagramBase.)
	OnNodeSelected	Raises the NodeSelected event. (Inherited from DiagramBase.)
	OnNodeSelecting	Raises the NodeSelecting event. (Inherited from DiagramBase.)
	OnNodeStartModifying	Raises the NodeStartModifying event. (Inherited from DiagramBase.)
	OnNodeTextEdited	Raises the **NodeTextEdited** event. (Inherited from DiagramBase.)
	OnNodeTextEditing	Raises the NodeTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnOriginAnchorChanged	Raises the OriginAnchorChanged event. (Inherited from DiagramBase.)

	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from DiagramBase.)
	OnQueryAlignTarget	Raises the QueryAlignTarget event. (Inherited from DiagramBase.)
	OnRepaint	Raises the Repaint event. (Inherited from DiagramBase.)
	OnRequestClipboard	Raises the RequestClipboard event. (Inherited from DiagramBase.)
	OnRequestDrawInteractions	Raises the RequestDrawInteractions event. (Inherited from DiagramBase.)
	OnRequestTimer	Raises the RequestTimer event. (Inherited from DiagramBase.)
	OnSelectionChanged	Raises the SelectionChanged event. (Inherited from DiagramBase.)
	OnSelectionModified	Raises the SelectionModified event. (Inherited from DiagramBase.)
	OnSelectionModifying	Raises the SelectionModifying event. (Inherited from DiagramBase.)
	OnSerializeTag	Raises the SerializeTag event. (Inherited from DiagramBase.)
	OnSetSelfLoopShape	Raises the SetSelfLoopShape event. (Inherited from DiagramBase.)
	OnShapeModified	Raises the ShapeModified event. (Inherited from DiagramBase.)
	OnShapeNotDefined	Raises the ShapeNotDefined event. (Inherited from DiagramBase.)
	OnTableColumnResized	Raises the TableColumnResized event. (Inherited from DiagramBase.)
	OnTableColumnResizing	Raises the TableColumnResizing event. (Inherited from DiagramBase.)
	OnTableRowResized	Raises the TableRowResized event. (Inherited from DiagramBase.)
	OnTableRowResizing	Raises the TableRowResizing event. (Inherited from DiagramBase.)
	OnTableSectionCollapsed	Raises the TableSectionCollapsed event. (Inherited from DiagramBase.)
	OnTableSectionExpanded	Raises the TableSectionExpanded event. (Inherited from DiagramBase.)
	OnTreeCollapsed	Raises the TreeCollapsed event. (Inherited from DiagramBase.)
	OnTreeExpanded	Raises the TreeExpanded event. (Inherited from DiagramBase.)
	OnTreeItemTextEditing	Raises the **TreeItemTextEditing** event. (Inherited from DiagramBase.)
	OnValidateAnchorPoint	Raises the ValidateAnchorPoint event. (Inherited from DiagramBase.)
	OnValidateLinkCrossing	Raises the ValidateLinkCrossing event. (Inherited from DiagramBase.)
	OnViewRemoved	Raises the **ViewRemoved** event. (Inherited from DiagramBase.)
	RemoveItem	Invoked by the **AddItemCmd** and **RemoveItemCmd** classes to effectively remove the specified item from the diagram. (Inherited from Diagram.)

Public Events

	Name	Description
	ActionRecorded	Raised when an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRecording	A validation event raised before an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRedone	Raised when an action is redone. (Inherited from DiagramBase.)
	ActionUndone	Raised when an action is undone. (Inherited from DiagramBase.)
	ActiveLayerChanged	Raised when the diagram's active layer has changed. (Inherited from DiagramBase.)
	BeginLoad	Raised when the component starts loading diagram data from a file or memory stream. (Inherited from DiagramBase.)

	BoundsChanged	Raised when the **Bounds** property value has changed. (Inherited from DiagramBase.)
	CellClicked	Raised when the user clicks a table's cell. (Inherited from DiagramBase.)
	CellDoubleClicked	Raised when the user double-clicks a table's cell. (Inherited from DiagramBase.)
	CellTextEdited	Raised when the user edits the text of a table's cell. (Inherited from DiagramBase.)
	CellTextEditing	A validation event that lets you approve or reject changes made to the text of a table's cell. (Inherited from DiagramBase.)
	Clicked	Raised when the user clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	ContainerChildAdded	Raised when a node is added to a ContainerNode, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerChildRemoved	Raised when a node is removed from a **ContainerNode**, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerFolded	Raised when the user folds a container by clicking the arrow-up button in the container's caption bar. (Inherited from DiagramBase.)
	ContainerPageChanged	Raised when the user changes current page of a PagedContainerNode. (Inherited from DiagramBase.)
	ContainerUnfolded	Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar. (Inherited from DiagramBase.)
	ContentVisibilityChanged	Raised when the visibility of container's content changes. (Inherited from DiagramBase.)
	DefaultShapeChanged	Raised when the **DefaultShape** property value has changed. (Inherited from DiagramBase.)
	DeserializeTag	Raised when the Tag of a diagram item must be deserialized from XML format. (Inherited from DiagramBase.)
	DestinationAnchorChanged	Raised when the DestinationAnchor property value of a link has changed. (Inherited from DiagramBase.)
	DirtyChanged	Raised when the **Dirty** property value has changed. (Inherited from DiagramBase.)
	DoubleClicked	Raised when the user double-clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	DrawAdjustmentHandles	Raised when item's adjustment handles must be custom-drawn. (Inherited from DiagramBase.)
	DrawAnchorPoint	Raised when an anchor point must be custom-drawn. (Inherited from DiagramBase.)
	DrawBackground	Raised to let you custom draw the diagram background. (Inherited from DiagramBase.)
	DrawBranchIndicator	Raised when a branch indicator must be custom-drawn. (Inherited from DiagramBase.)
	DrawCell	Raised when a table's cell must be custom drawn. (Inherited from DiagramBase.)
	DrawForeground	Raised to let you custom draw the diagram foreground. (Inherited from DiagramBase.)
	DrawLink	Raised when a diagram link must be custom drawn. (Inherited from DiagramBase.)
	DrawNode	Raised when a diagram node must be custom drawn. (Inherited from DiagramBase.)

	EndLoad	Raised when the component ends loading diagram data from a file or memory stream. (Inherited from DiagramBase.)
	ExpandButtonClicked	Raised when the user clicks the [±] button of an Expandable node. (Inherited from DiagramBase.)
	GroupDestroyed	Raised when a Group object is destroyed. (Inherited from DiagramBase.)
	HitTestAdjustmentHandles	Raised when custom hit testing of item's adjustment handles must be performed. (Inherited from DiagramBase.)
	InitializeLasso	Raised when the user starts drawing a selection lasso, before it is rendered on screen. (Inherited from DiagramBase.)
	InitializeLink	Raised when the user starts drawing a new link, just after the **DiagramLink** instance is created. (Inherited from DiagramBase.)
	InitializeNode	Raised when the user starts drawing a new node, just after the **DiagramNode** instance is created. (Inherited from DiagramBase.)
	ItemAdded	Raised when an item is added to the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	ItemRemoved	Raised when an item is removed from the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	JsonDeserializeTag	Raised when the **Tag** of a diagram item must be deserialized from JSON. (Inherited from DiagramBase.)
	JsonSerializeTag	Raised when the **Tag** of a diagram item must be serialized to JSON. (Inherited from DiagramBase.)
	LayerVisibilityChanged	Raised when the visibility of a Layer changes. (Inherited from DiagramBase.)
	LinkActivated	Raised when a link becomes the **ActiveItem**. (Inherited from DiagramBase.)
	LinkClicked	Raised when the user clicks a link. (Inherited from DiagramBase.)
	LinkCreateCancelled	Raised when the user cancels link creation. (Inherited from DiagramBase.)
	LinkCreated	Raised when the user draws a new link. (Inherited from DiagramBase.)
	LinkCreating	A validation event raised while the user is drawing a new link. (Inherited from DiagramBase.)
	LinkDeactivated	Raised when a link is no longer the **ActiveItem**. (Inherited from DiagramBase.)
	LinkDeleted	Raised when a link is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDeleting	Raised when the user tries to delete a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkDeselected	Raised when a link is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDoubleClicked	Raised when the user double-clicks a link. (Inherited from DiagramBase.)
	LinkHovered	Raised when the mouse pointer hovers over a link. (Inherited from DiagramBase.)
	LinkModified	Raised when the user moves any control point of a link. (Inherited from DiagramBase.)
	LinkModifyCancelled	Raised when the user cancels move or resize operation being done on a link. (Inherited from DiagramBase.)
	LinkModifying	A validation event raised while the user is modifying a link. (Inherited from DiagramBase.)

	LinkPasted	Raised when a link has been pasted from the clipboard. (Inherited from DiagramBase.)
	LinkRouted	Raised when a link is routed by the auto-routing algorithm. (Inherited from DiagramBase.)
	LinkSelected	Raised when a link is selected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkSelecting	Raised when the user tries to select a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkSplit	Raised when the user splits a link interactively. (Inherited from DiagramBase.)
	LinkStartModifying	A validation event raised when the user starts modifying a link. (Inherited from DiagramBase.)
	LinkTextEdited	Raised when the user edits the text of a link. (Inherited from DiagramBase.)
	LinkTextEditing	A validation event that lets you approve or reject changes made to the text of a link. (Inherited from DiagramBase.)
	MeasureUnitChanged	Raised when the **MeasureUnit** property value has changed. (Inherited from DiagramBase.)
	NodeActivated	Raised when a node becomes the **ActiveItem**. (Inherited from DiagramBase.)
	NodeClicked	Raised when the user clicks a node. (Inherited from DiagramBase.)
	NodeCreateCancelled	Raised when the user cancels node creation. (Inherited from DiagramBase.)
	NodeCreated	Raised when the user draws a new node. (Inherited from DiagramBase.)
	NodeCreating	A validation event raised while the user is drawing a new node. (Inherited from DiagramBase.)
	NodeDeactivated	Raised when a node is no longer the **ActiveItem**. (Inherited from DiagramBase.)
	NodeDeleted	Raised when a node is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDeleting	Raised when the user tries to delete a node, this event lets you cancel the operation. (Inherited from DiagramBase.)
	NodeDeselected	Raised when a node is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDoubleClicked	Raised when the user double-clicks a node. (Inherited from DiagramBase.)
	NodeHovered	Raised when the mouse pointer hovers over a node. (Inherited from DiagramBase.)
	NodeModified	Raised when the user moves or resizes a node. (Inherited from DiagramBase.)
	NodeModifyCancelled	Raised when the user cancels move or resize operation being done on a node. (Inherited from DiagramBase.)
	NodeModifying	A validation event raised while the user is moving or resizing a node. (Inherited from DiagramBase.)
	NodePasted	Raised when a node has been pasted from the clipboard. (Inherited from DiagramBase.)
	NodeSelected	Raised when a node is selected, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeSelecting	Raised when the user tries to select a node, this event lets you cancel the operation. (Inherited from DiagramBase.)

	NodeStartModifying	A validation event raised when the user starts modifying a node. (Inherited from DiagramBase.)
	NodeTextEdited	Raised when the user edits the text of a node. (Inherited from DiagramBase.)
	NodeTextEditing	A validation event that lets you approve or reject changes made to the text of a node. (Inherited from DiagramBase.)
	OriginAnchorChanged	Raised when the OriginAnchor property value of a link has changed. (Inherited from DiagramBase.)
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation. (Inherited from DiagramBase.)
	QueryAlignTarget	Raised to validate a node as target for alignment guides when **AutoAlignNodes** is enabled. (Inherited from DiagramBase.)
	Repaint	Raised when the diagram views should be repainted. (Inherited from DiagramBase.)
	RequestCacheReset	Raised when an object within the diagram requests a reset of the cached bitmap. (Inherited from DiagramBase.)
	RequestClipboard	Raised when an object within the diagram requests an operation with the clipboard. (Inherited from DiagramBase.)
	RequestDrawInteractions	For internal use. (Inherited from DiagramBase.)
	RequestTimer	Raised when an object within the diagram needs to start or stop a timer. (Inherited from DiagramBase.)
	SelectionChanged	Raised when an item is added to or removed from the selection (Inherited from DiagramBase.)
	SelectionModified	Raised when the user moves multiple selected items. (Inherited from DiagramBase.)
	SelectionModifying	Raised while the user is moving multiple selected items, this event lets you cancel the modification. (Inherited from DiagramBase.)
	SerializeTag	Raised when the **Tag** of a diagram item must be serialized into an XML document. (Inherited from DiagramBase.)
	SetSelfLoopShape	Raised when a link becomes a self-loop, giving you a chance to override the default look shape. (Inherited from DiagramBase.)
	ShapeModified	Raised when the user modifies the shape of a ShapeNode by moving one of the control points corresponding to shape formula parameters. (Inherited from DiagramBase.)
	ShapeNotDefined	Raised when the diagram being loaded contains nodes with an undefined shape. (Inherited from DiagramBase.)
	TableColumnResized	Raised when the user finishes resizing a column in a table. (Inherited from DiagramBase.)
	TableColumnResizing	Raised when the user resizes a column in a table interactively. (Inherited from DiagramBase.)
	TableRowResized	Raised when the user finishes resizing a row in a table. (Inherited from DiagramBase.)
	TableRowResizing	Raised when the user resizes a row in a table interactively. (Inherited from DiagramBase.)
	TableSectionCollapsed	Raised when the user collapses the rows of a table's section by clicking the [-] button in the section's header row. (Inherited from DiagramBase.)
	TableSectionExpanded	Raised when the user expands the rows of a table's section by clicking the [+] button in the section's header row. (Inherited from DiagramBase.)
	TreeCollapsed	Raised when a user collapses a tree branch by clicking the [-] button of an **Expandable** node. (Inherited from DiagramBase.)

	TreeExpanded	Raised when a user expands a tree branch by clicking the [+] button of an **Expandable** node. (Inherited from DiagramBase.)
	TreeItemTextEditing	A validation event that lets you approve or reject changes made to the text of a TreeViewNode's item. (Inherited from DiagramBase.)
	ValidateAnchorPoint	A validation event that lets you approve or reject connecting a link to an anchor point. (Inherited from DiagramBase.)
	ValidateLinkCrossing	A validation event that lets you approve or reject displaying a decoration at the intersection point of links. (Inherited from DiagramBase.)
	ViewRemoved	Raised when a DiagramView is detached from this diagram. (Inherited from DiagramBase.)

See Also

DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage Constructor

Initializes a new instance of the DiagramPage class.

Namespace: MindFusion.Diagramming
Assembly: DiagramPage.js

Syntax

JavaScript Copy Code

```
function DiagramPage ([title])
```

Parameters

title
Optional.
String. The page title.

See Also

DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage Properties

DiagramPage.iconUrl Property

Gets or sets URL of the image to paint as the page's icon.

Namespace: MindFusion.Diagramming
Assembly: DiagramPage.js

Syntax

JavaScript Copy Code

```
get iconUrl() {}
set iconUrl(value) {}
```

Property Value

String

A string containing image URL.

See Also

DiagramPage Members
DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage.title Property

Gets or sets the page title.

Namespace: MindFusion.Diagramming
Assembly: DiagramPage.js

Syntax

JavaScript Copy Code

```
get title() {}
set title(value) {}
```

Property Value

String

A string specifying the page title.

See Also

DiagramPage Members
DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage.titleColor Property

Gets or sets the color of the title when the title is displayed.

Namespace: MindFusion.Diagramming
Assembly: DiagramPage.js

Syntax

JavaScript Copy Code

```
get titleColor() {}
set titleColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

DiagramPage Members
DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage Events

DiagramView Class

Represents a control that displays the content of a Diagram object.

Syntax

JavaScript — Copy Code

```
// class
DiagramView.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.CanvasControl
  MindFusion.Diagramming.DiagramView

See Also

DiagramView Members
MindFusion.Diagramming Namespace

DiagramView Members

Represents a control that displays the content of a Diagram object.

The following tables list the members exposed by the DiagramView type.

Public Constructors

	Name	Description
	DiagramView	Initializes a new instance of the DiagramView class.

Public Properties

	Name	Description
	allowInplaceEdit	Gets or sets whether users are allowed to edit the text of diagram items.
	autoScroll	Gets or sets a value indicating whether automatic scrolling is enabled.
	autoScrollAmount	Gets or sets the amount by which to auto-scroll the view when the mouse leaves the view boundaries while drawing.
	behavior	Gets or sets a value indicating how the control responds to user actions.
	defaultControlTemplate	Gets or sets HTML string representing the DOM content rendered by default by newly created ControlNode objects.
	delKeyAction	Gets or sets what action to perform when users hit the Del key.
	diagram	Gets or sets the diagram displayed by this view.
	dragIndicatorOpacity	Gets or sets the opacity of ghost image displayed during drag-and-drop from NodeListView.
	enableMultiTouch	Gets or sets a value indicating whether the control handles DOM Pointer events to implement multi-touch interactions, such as zooming, node rotation or simultaneous drawing of multiple diagram items.
	handleTouchHitDistance	Gets or sets maximum distance between touch position and adjustment handle at which the handle can still be grabbed to modify its item.
	horizontalDropAlignment	Gets or sets the horizontal placement of nodes created by drag-and-drop operations, relatively to mouse cursor.
	hoverDelay	Gets or sets the time in milliseconds that mouse pointer should remain stationary before reporting hover events.
	inplaceEditAcceptOnEnter	Gets or sets a value indicating whether pressing the ENTER keyshould exit inplace-edit mode and accept changes made to text.

	inplaceEditCancelOnEsc	Gets or sets a value indicating whether pressing the ESC keyshould exit inplace-edit mode and reject changes made to text.
	leftButtonActions	Gets or sets a combination of flags that specify what actions can be performed via the left mouse button.
	licenseLocation	Gets or sets URL of the control's license file.
	magnifierEnabled	Gets or sets a value indicating whether the magnifier tool is currently enabled.
	magnifierFactor	Gets or sets the zoom factor of the magnifier tool.
	magnifierFrameColor	Gets or sets the color of the magnifier tool's frame.
	magnifierFrameThickness	Gets or sets the thickness of the magnifier frame.
	magnifierHeight	Gets or sets the height of the magnifier tool.
	magnifierSecondaryFrameColor	Gets or sets the color of the magnifier tool's secondary frame.
	magnifierShading	Gets or sets whether to enhance the effect of a magnifier lense by using a gradient shading.
	magnifierShape	Gets or sets a reference to the geometric shape definition of the magnifier tool.
	magnifierWidth	Gets or sets the width of the magnifier tool.
	middleButtonActions	Gets or sets a combination of flags that specify what actions can be performed via the middle mouse button.
	modificationStart	Gets or sets how users can start modifying diagram items.
	modifierKeyActions	Gets an object that allows changing the function of keyboard modifier keys such as Ctrl and Alt.
	mouseWheelAction	Gets or sets what action to perform when users rotate the mouse wheel.
	multiTouchDraw	Gets or sets a value indicating whether the control allows drawing multiple items simultaneously.
	multiTouchModify	Gets or sets a value indicating whether multi-touch modifications are enabled.
	multiTouchZoom	Gets or sets a value indicating whether multi-touch zoom is enabled.
	rightButtonActions	Gets or sets a combination of flags that specify what actions can be performed via the right mouse button.
	scrollX	Gets or sets the diagram's horizontal scroll position.
	scrollY	Gets or sets the diagram's vertical scroll position.
	scrollZoneSize	Gets or sets the size of the auto scroll zone near the edges of the control.
	tooltipDelay	Gets or sets the delay before a tooltip is shown.
	verticalDropAlignment	Gets or sets the vertical placement of nodes created by drag-and-drop operations, relatively to mouse cursor.
	viewport	Returns the visible portion of the diagram area.
	virtualScroll	Gets or sets a value indicating whether virtual scroll mode is enabled.
	zoomFactor	Gets or sets the zoom factor.

Public Methods

	Name	Description
	animate	Animates the view state.

	beginEdit	Begins in-place editing of the specified object.
	bringIntoView	Scrolls the view to make the specified item visible.
	bringRectIntoView	Scrolls the view to make the specified rectangular area visible.
	cancelDrag	Cancels all current drag operations.
	clearTooltip	Forces the removal of any created tooltips from the diagram canvas.
	copyToClipboard	Copies the current selection of items to the clipboard.
S	create	Creates and initializes a new **DiagramView** from the specified element. This method is static and can be called without creating an instance of the class.
	cutToClipboard	Cuts the currently selected items to the clipboard.
	dispose	Control.dispose override.
	endEdit	Exits the in-place editing mode and either accepts or rejects the changes made to the item's text.
S	find	Returns the specified **DiagramView** object. This member is static and can be invoked without creating an instance of the class.
	fromJson	Loads the diagram from a JSON string.
	initialize	Control.initialize override.
	loadFromXml	Loads the diagram from an XML file.
	pasteFromClipboard	Pastes diagram items from the clipboard.
	print	Exports the diagram as a list of images to a new HTML document and opens the browser's print dialog.
	printPreview	Exports the diagram as a list of images to a new HTML document.
	record	Starts recording mouse and keyboard events for subsequent replay.
	replay	Replays a sequence of recorded mouse and keyboard events created by record and stopRecording methods.
	saveToXml	Saves the diagram to an XML file.
	scrollTo	Scrolls the diagram to the specified point.
	setScroll	IZoomTarget implementation
	setZoomFactorPivot	Sets the zoom factor, preserving the on-screen position of the specified diagram point.
	stopRecording	Stops recording mouse and keyboard events.
	toJson	Saves the diagram to a JSON string.
	zoomToFit	Zooms the view to fit the **Diagram** contents in component's area.
	zoomToRect	Zooms and scrolls the view to fit the specified **Diagram** region in component's area.

Public Events

	Name	Description
	createEditControl	Raised to let you create a custom inplace editor control.
	diagramChanged	Raised when the **diagram** property changes.
	enterInplaceEditMode	Raised when entering inplace-edit mode.

	leaveInplaceEditMode	Raised when leaving inplace-edit mode.
	mouseWheel	Raised when the user rotates the mouse wheel while the cursor is positioned over the diagram surface.
	zoomChanged	Raised when the zoom factor property changes.

See Also

DiagramView Class
MindFusion.Diagramming Namespace

DiagramView Constructor

Initializes a new instance of the DiagramView class.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function DiagramView (element, [diagram])
```

Parameters

element
HTMLCanvasElement. The Canvas DOM Element this DiagramView is associated with.

diagram
Optional.
Diagram. The Diagram instance this control is associated with.

See Also

DiagramView Class
MindFusion.Diagramming Namespace

DiagramView Methods

DiagramView.animate Method

Animates the view state.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function animate (animation)
```

Parameters

animation
ViewAnimation. A **ViewAnimation** -derived object.

Example

This example centers specified node in the viewport by running a ScrollAnimation:

JavaScript Copy Code

```
var endNode = diagram.nodes.find(
    n => n.id == "end");

if (endNode != null)
{
    var viewSize = diagramView.viewport.getSize();
    var newScroll = new Point(
        endNode.center.x - viewSize.width / 2,
        endNode.center.y - viewSize.height / 2);

    var currentScroll = new Point(
        diagramView.scrollX, diagramView.scrollY);

    diagramView.animate(
        new ScrollAnimation(
        {
            fromValue: currentScroll,
            toValue: newScroll,
            animationType: Animations.AnimationType.Power,
            easingType: Animations.EasingType.EaseInOut
        }));
}
```

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.beginEdit Method

Begins in-place editing of the specified object.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function beginEdit (item, [point, [editTarget]])
```

Parameters

item
InplaceEditable. The DiagramItem whose text should be edited.

point
Optional.
Point. A Point specifying where the in-place edit box should appear.

editTarget
Optional.
Cell | LinkLabel. The element to use as a target for the inplace edit.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.bringIntoView Method

Scrolls the view to make the specified item visible.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function bringIntoView (item)
```

Parameters

item
DiagramItem. The item to bring into view.

Remarks

Scrolls the diagram view to ensure that the specified item is visible. No scrolling is done if the item is already inside the viewport.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
scrollTo Method
bringRectIntoView Method
scrollX Property
scrollY Property
zoomToRect Method

DiagramView.bringRectIntoView Method

Scrolls the view to make the specified rectangular area visible.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function bringRectIntoView (rect)
```

Parameters

rect
Rect. A Rect specifying the diagram region to show in the viewport.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.cancelDrag Method

Cancels all current drag operations.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function cancelDrag ()
```

Remarks

This method is useful in situations when the current draw operations must be cancelled, for example when starting external drag-and-drop from **pointerMove** DOM event handlers, or in response to Esc keypress. Multiple operations could be running

simultaneously when enableMultiTouch is set to **true**, and this method cancels them all. Individual operations can be cancelled by calling cancelDrag method of event arguments when handling validation events, such as nodeCreating.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.clearTooltip Method

Forces the removal of any created tooltips from the diagram canvas.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function clearTooltip ()
```

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.copyToClipboard Method

Copies the current selection of items to the clipboard.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function copyToClipboard ([systemClipboard])
```

Parameters

systemClipboard
Optional.
Boolean. **true** to copy to the system clipboard, or **false** otherwise. The default value is **false**.

Remarks

The diagram control maintains its own local clipboard and items are copied to it by default. If you specify a **true** argument and running under browsers that allow access to the system clipboard, the items are copied to the system clipboard; in all other cases they are copied to the local clipboard.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
cutToClipboard Method
pasteFromClipboard Method

DiagramView.create Method

Creates and initializes a new DiagramView from the specified element. This method is static and can be called without creating an instance of the class.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
function create (element, [diagram])
```

Parameters

element
HTMLCanvasElement. The DOM <canvas> element that the DiagramView should be attached to.

diagram
Optional.
Diagram. The Diagram instance that should be rendered by the new view.

Return Value

DiagramView. The newly created **DiagramView** object.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
find Method
WebComponents.register Method

DiagramView.cutToClipboard Method

Cuts the currently selected items to the clipboard.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
function cutToClipboard ([systemClipboard])
```

Parameters

systemClipboard
Optional.
Boolean. **true** to copy to the system clipboard, or **false** otherwise. The default value is **false**.

Remarks

The diagram control maintains its own local clipboard and items are copied to it by default. If you specify a **true** argument and running under browsers that allow access to the system clipboard, the items are copied to the system clipboard; in all other cases they are copied to the local clipboard.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
copyToClipboard Method
pasteFromClipboard Method

DiagramView.dispose Method

Control.dispose override.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function dispose ()
```

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.endEdit Method

Exits the in-place editing mode and either accepts or rejects the changes made to the item's text.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function endEdit ([accept])
```

Parameters

accept
Optional.
Boolean. **true** to accept changes made to text, or **false** to reject them.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.find Method

Returns the specified DiagramView object. This member is static and can be invoked without creating an instance of the class.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function find (id)
```

Parameters

id
String. The id of the DiagramView's element.

Return Value

DiagramView. A **DiagramView** object with the specified id, if found, or **null** otherwise.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
create Method

DiagramView.fromJson Method

Loads the diagram from a JSON string.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
function fromJson ()
```

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.initialize Method

Control.initialize override.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
function initialize ()
```

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.loadFromXml Method

Loads the diagram from an XML file.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
function loadFromXml (fileUrl, [onLoad, [onError]])
```

Parameters

fileUrl
String. The URL of an XML file where the data should be read from.

onLoad
Optional.
function. A callback that should be invoked if the file is loaded successfully.

onError
Optional.
function. A callback that should be invoked if the file could not be downloaded.

See Also

DiagramView.pasteFromClipboard Method

Pastes diagram items from the clipboard.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function pasteFromClipboard (offsetX, offsetY, [systemClipboard, [unconnectedLinks]])
```

Parameters

offsetX
Number. Horizontal offset of pasted items from their original positions.

offsetY
Number. Vertical offset of pasted items from their original positions.

systemClipboard
Optional.
Boolean. **true** to paste from the system clipboard, or **false** otherwise. The default value is **false**.

unconnectedLinks
Optional.
Boolean. **true** to allow pasting links whose origin or destination node has not been copied, or **false** otherwise. The default value is **false**.

Remarks

The diagram control maintains its own local clipboard and items are copied to it by default. If you specify **true** for the *systemClipboard* argument and running under browsers that allow access to the system clipboard, the items are copied to the system clipboard; in all other cases they are copied to the local clipboard.

Due to **Navigator.clipboard** API being asynchronous, pasted items are not available immediately after calling this method with *systemClipboard* enabled. You will be able to access them only after nodePasted and linkPasted events are raised.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
copyToClipboard Method
cutToClipboard Method

DiagramView.print Method

Exports the diagram as a list of images to a new HTML document and opens the browser's print dialog.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function print ([options])
```

Parameters

options
Optional.
Object. An object, containing customization properties, including printArea (defaults to diagram.getContentBounds()) and pageSize (defaults to diagramView.viewport).

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
printPreview Method

DiagramView.printPreview Method

Exports the diagram as a list of images to a new HTML document.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function printPreview ([options])
```

Parameters

options
Optional.
Object. An object, containing customization properties, including printArea (defaults to diagram.getContentBounds()) and pageSize (defaults to diagramView.viewport).

Example

JavaScript Copy Code

```
document.getElementById("printPreview").addEventListener("click", () => {
    var diagramView = DiagramView.find("diagram");

    // preferred image size is A4 with subtracted margins
    var A4 = new Size(210, 297);
    A4.width -= 25.4;
    A4.height -= 25.4;

    diagramView.printPreview(
        {
            printArea: diagramView.diagram.bounds,
            pageSize: A4
        });
});
```

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
print Method

DiagramView.record Method

Starts recording mouse and keyboard events for subsequent replay.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function record ()
```

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
replay Method
stopRecording Method

DiagramView.replay Method

Replays a sequence of recorded mouse and keyboard events created by record and stopRecording methods.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function replay (recording, timer, [savedState, [timerInterval]])
```

Parameters

recording
Array. An array of event records returned by stopRecording.

timer
Boolean. **true** to replay records using a timer, preserving their original relative timing, or **false** to replay them immediately.

savedState
Optional.
String. A diagram JSON string that should be loaded before replaying recorded events.

timerInterval
Optional.
Number. A number of milliseconds to use as a constant interval instead of relative timing.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
record Method
stopRecording Method

DiagramView.saveToXml Method

Saves the diagram to an XML file.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function saveToXml (url)
```

Parameters

url
String. A URL specifying where the diagram's XML should be posted to.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.scrollTo Method

Scrolls the diagram to the specified point.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
function scrollTo (x, y)
```

Parameters

x
Number. A number specifying the new horizontal scroll position in the current unit of measure.

y
Number. A number specifying the new vertical scroll position in the current unit of measure.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
bringIntoView Method

DiagramView.setScroll Method

IZoomTarget implementation

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
function setScroll ()
```

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.setZoomFactorPivot Method

Sets the zoom factor, preserving the on-screen position of the specified diagram point.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
function setZoomFactorPivot (zoomFactor, pivotPoint)
```

Parameters

zoomFactor
Number. The new zoom factor.

pivotPoint
Point. A Point instance specifying the zoom center.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.stopRecording Method

Stops recording mouse and keyboard events.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function stopRecording ()
```

Return Value

Array. An array of event records representing user's input.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
record Method
replay Method

DiagramView.toJson Method

Saves the diagram to a JSON string.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

Return Value

String. A string containing the diagram's JSON representation.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.zoomToFit Method

Zooms the view to fit the Diagram contents in component's area.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function zoomToFit ()
```

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.zoomToRect Method

Zooms and scrolls the view to fit the specified Diagram region in component's area.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
function zoomToRect (rect)
```

Parameters

rect
Rect. A Rect instance specifying desired viewport position and size.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
zoomFactor Property
bringRectIntoView Method
bringIntoView Method

DiagramView Properties

DiagramView.allowInplaceEdit Property

Gets or sets whether users are allowed to edit the text of diagram items.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get allowInplaceEdit() {}
set allowInplaceEdit(value) {}
```

Property Value

Boolean

true to allow in-place editing, or **false** otherwise. The default value is **false**.

Remarks

Specifies whether users are allowed to edit text of diagram nodes in-place. If enabled, a text-entry field appears at node's position when the node is double-clicked. In-place editing mode can be exited by clicking anywhere outside the text field.

To enter or exit in-place editing mode programmatically, call the beginEdit and endEdit methods of DiagramView.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
inplaceEditAcceptOnEnter Property

inplaceEditCancelOnEsc Property
nodeTextEdited Event
linkTextEdited Event
enterInplaceEditMode Event

DiagramView.autoScroll Property

Gets or sets a value indicating whether automatic scrolling is enabled.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get autoScroll() {}
set autoScroll(value) {}
```

Property Value

Boolean

true to enable auto scrolling, or **false** otherwise. The default value is **true**.

Remarks

If enabled, the control will scroll its content automatically when the user drags the mouse near the viewport edges. This works only if the diagram view is placed inside a DIV whose **overflow** attribute is set to **scroll** or **auto**.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
scrollZoneSize Property
scrollX Property
scrollY Property
bringIntoView Method

DiagramView.autoScrollAmount Property

Gets or sets the amount by which to auto-scroll the view when the mouse leaves the view boundaries while drawing.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get autoScrollAmount() {}
set autoScrollAmount(value) {}
```

Property Value

Number

A number specifying the amount by which to auto-scroll.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.behavior Property

Gets or sets a value indicating how the control responds to user actions.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get behavior() {}
set behavior(value) {}
```

Property Value

Behavior

A member of the Behavior enumeration. The default value is **LinkShapes**.

Remarks

The property specifies how the control responds to user actions with the mouse. The default value, LinkShapes, facilitates drawing of flowchart diagrams. **LinkTables** makes JsDiagram create table nodes and links while the user draws with the mouse, allowing creation of entity-relationship or class-hierarchy diagrams.

Modes such as **Modify**, **PanAndModify**, **MoveNodes**, **SelectOnly** allow users to interact with existing items, but not draw new ones on the canvas. Note that NodeListView can be used to create new nodes regardless of DiagramView's current behavior mode.

There are several more values that can be assigned to the property, that make the control draw specific types of items, for example only shape nodes, only free-form nodes, only links, and so on. These are useful when implementing the ability to select specific drawing tools in applications.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.defaultControlTemplate Property

Gets or sets HTML string representing the DOM content rendered by default by newly created ControlNode objects.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get defaultControlTemplate() {}
set defaultControlTemplate(value) {}
```

Property Value

String

An HTML string.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.delKeyAction Property

Gets or sets what action to perform when users hit the Del key.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get delKeyAction() {}
set delKeyAction(value) {}
```

Property Value

DelKeyAction

A member of the DelKeyAction enumeration. The default value is **DeleteSelectedItems**.

Remarks

Depending on this property value, the control can delete all selected items, delete only the active item, or ignore the Del key press.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
nodeDeleting Event
linkDeleting Event
autoDeleteLinks Property
showDeleteButton Property

DiagramView.diagram Property

Gets or sets the diagram displayed by this view.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get diagram() {}
set diagram(value) {}
```

Property Value

Diagram

The Diagram instance displayed by this view.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.dragIndicatorOpacity Property

Gets or sets the opacity of ghost image displayed during drag-and-drop from NodeListView.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get dragIndicatorOpacity() {}
set dragIndicatorOpacity(value) {}
```

Property Value

Number

A number specifying opacity. The default value is 0.4.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.enableMultiTouch Property

Gets or sets a value indicating whether the control handles DOM Pointer events to implement multi-touch interactions, such as zooming, node rotation or simultaneous drawing of multiple diagram items.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get enableMultiTouch() {}
set enableMultiTouch(value) {}
```

Property Value

Boolean

true to enable multi-touch, or **false** otherwise. The default value is **true**.

Remarks

Set this property to **false** if you need to handle touch events yourself or through a third-party gesture library. If disabled, the control will revert to its mouse event handling code from previous versions. It will also no longer be able to capture the mouse input during drag operations (to detect mouse pointer leaving the view), which is handled through DOM Pointers API.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
multiTouchModify Property
multiTouchZoom Property
multiTouchDraw Property

DiagramView.handleTouchHitDistance Property

Gets or sets maximum distance between touch position and adjustment handle at which the handle can still be grabbed to modify its item.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get handleTouchHitDistance() {}
set handleTouchHitDistance(value) {}
```

Property Value

Number

A number specifying the allowed hit distance, in diagram's current unit. The default value is 5.

Remarks

This value is considered only when handling touch events.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.horizontalDropAlignment Property

Gets or sets the horizontal placement of nodes created by drag-and-drop operations, relatively to mouse cursor.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get horizontalDropAlignment() {}
set horizontalDropAlignment(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration. The default value is **Near**.

Remarks

This property lets you center the new node or align specified edge to the mouse pointer position when drag-and-drop ends.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
verticalDropAlignment Property

DiagramView.hoverDelay Property

Gets or sets the time in milliseconds that mouse pointer should remain stationary before reporting hover events.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get hoverDelay() {}
set hoverDelay(value) {}
```

Property Value

Number

The hover delay in milliseconds. The default value is 1500.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
nodeHovered Event
linkHovered Event

DiagramView.inplaceEditAcceptOnEnter Property

Gets or sets a value indicating whether pressing the ENTER keyshould exit inplace-edit mode and accept changes made to text.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get inplaceEditAcceptOnEnter() {}
set inplaceEditAcceptOnEnter(value) {}
```

Property Value

Boolean

true to end inplace editing and accept changes when ENTER is pressed, or **false** otherwise. The default value is **false**.

Remarks

If the value of this property is **false**, pressing the ENTER key inserts a new line in the text while in inplace-edit mode.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
allowInplaceEdit Property
inplaceEditCancelOnEsc Property

DiagramView.inplaceEditCancelOnEsc Property

Gets or sets a value indicating whether pressing the ESC keyshould exit inplace-edit mode and reject changes made to text.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get inplaceEditCancelOnEsc() {}
set inplaceEditCancelOnEsc(value) {}
```

Property Value

Boolean

true to end inplace editing and cancel changes when ESC is pressed, or **false** otherwise. The default value is **true**.

Remarks

If the value of this property is **false**, pressing the ESC key has no effect while in inplace-edit mode.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
allowInplaceEdit Property
inplaceEditAcceptOnEnter Property

DiagramView.leftButtonActions Property

Gets or sets a combination of flags that specify what actions can be performed via the left mouse button.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get leftButtonActions() {}
set leftButtonActions(value) {}
```

Property Value

MouseButtonActions

A combination of MouseButtonActions members.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.licenseLocation Property

Gets or sets URL of the control's license file.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get licenseLocation() {}
set licenseLocation(value) {}
```

Property Value

String

A URL string that specifies the location of the license key.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.magnifierEnabled Property

Gets or sets a value indicating whether the magnifier tool is currently enabled.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get magnifierEnabled() {}
set magnifierEnabled(value) {}
```

Property Value

Boolean

true if the magnifier is enabled, or **false** otherwise. The default value is **false**.

Remarks

The magnifier tool can be alternatively enabled on-demand through a modifier key specified by modifierKeyActions, or a mouse button specified in leftButtonActions, middleButtonActions, rightButtonActions.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.magnifierFactor Property

Gets or sets the zoom factor of the magnifier tool.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get magnifierFactor() {}
set magnifierFactor(value) {}
```

Property Value

Number

The zoom factor of the magnifier tool as percentage.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.magnifierFrameColor Property

Gets or sets the color of the magnifier tool's frame.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get magnifierFrameColor() {}
set magnifierFrameColor(value) {}
```

Property Value

String

The color of the magnifier tool's frame.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.magnifierFrameThickness Property

Gets or sets the thickness of the magnifier frame.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get magnifierFrameThickness() {}
set magnifierFrameThickness(value) {}
```

Property Value

Number

A number specifying the thickness of the magnifier frame.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.magnifierHeight Property

Gets or sets the height of the magnifier tool.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get magnifierHeight() {}
set magnifierHeight(value) {}
```

Property Value

Number

A number specifying the height of the magnifier tool.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.magnifierSecondaryFrameColor Property

Gets or sets the color of the magnifier tool's secondary frame.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get magnifierSecondaryFrameColor() {}
set magnifierSecondaryFrameColor(value) {}
```

Property Value

String

The color of the magnifier tool's secondary frame.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.magnifierShading Property

Gets or sets whether to enhance the effect of a magnifier lense by using a gradient shading.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
get magnifierShading() {}
set magnifierShading(value) {}
```

Property Value

Boolean

true to enable shading, or **false** otherwise.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.magnifierShape Property

Gets or sets a reference to the geometric shape definition of the magnifier tool.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
get magnifierShape() {}
set magnifierShape(value) {}
```

Property Value

Shape | String

The Shape or shape identifier of the magnifier tool.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.magnifierWidth Property

Gets or sets the width of the magnifier tool.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
get magnifierWidth() {}
set magnifierWidth(value) {}
```

Property Value

Number

A number specifying the width of the magnifier tool.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.middleButtonActions Property

Gets or sets a combination of flags that specify what actions can be performed via the middle mouse button.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get middleButtonActions() {}
set middleButtonActions(value) {}
```

Property Value

MouseButtonActions

A combination of MouseButtonActions members.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.modificationStart Property

Gets or sets how users can start modifying diagram items.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get modificationStart() {}
set modificationStart(value) {}
```

Property Value

ModificationStart

A member of the ModificationStart enumeration.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.modifierKeyActions Property

Gets an object that allows changing the function of keyboard modifier keys such as Ctrl and Alt.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
get modifierKeyActions() {}
```

Property Value

ModifierKeyActions

An instance of the ModifierKeyActions class.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.mouseWheelAction Property

Gets or sets what action to perform when users rotate the mouse wheel.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
get mouseWheelAction() {}
set mouseWheelAction(value) {}
```

Property Value

MouseWheelAction

A member of the MouseWheelAction enumeration. The default value is **Scroll**.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
mouseWheel Event

DiagramView.multiTouchDraw Property

Gets or sets a value indicating whether the control allows drawing multiple items simultaneously.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
get multiTouchDraw() {}
set multiTouchDraw(value) {}
```

Property Value

Boolean

true to allow drawing multiple items simultaneously, or **false** otherwise. The default value is **true**.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

multiTouchModify Property
multiTouchZoom Property
enableMultiTouch Property

DiagramView.multiTouchModify Property

Gets or sets a value indicating whether multi-touch modifications are enabled.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get multiTouchModify() {}
set multiTouchModify(value) {}
```

Property Value

Boolean

true to enable multi-touch node modification, or **false** otherwise. The default value is **true**.

Remarks

If this property is enabled, two touches on same node will scale and rotate it. Otherwise each touch is handled independently according to current behavior value, e.g. by drawing a new diagram link.

Multi-touch rotation is applied only if the Rotate handle bit in enabledHandles is set.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
multiTouchZoom Property
multiTouchDraw Property
enableMultiTouch Property

DiagramView.multiTouchZoom Property

Gets or sets a value indicating whether multi-touch zoom is enabled.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get multiTouchZoom() {}
set multiTouchZoom(value) {}
```

Property Value

Boolean

true to enable multi-touch zoom, or **false** otherwise. The default value is **true**.

Remarks

If this property is disabled, each touch is handled independently according to current behavior value, e.g. by drawing a new diagram item or moving an existing one.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

multiTouchModify Property
enableMultiTouch Property

DiagramView.rightButtonActions Property

Gets or sets a combination of flags that specify what actions can be performed via the right mouse button.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get rightButtonActions() {}
set rightButtonActions(value) {}
```

Property Value

MouseButtonActions

A combination of MouseButtonActions members.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.scrollX Property

Gets or sets the diagram's horizontal scroll position.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get scrollX() {}
set scrollX(value) {}
```

Property Value

Number

A number specifying the horizontal scroll position in the current unit of measure.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
scrollY Property
autoScroll Property
scrollTo Method
bringIntoView Method

DiagramView.scrollY Property

Gets or sets the diagram's vertical scroll position.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get scrollY() {}
set scrollY(value) {}
```

Property Value

Number

A number specifying the vertical scroll position in the current unit of measure.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
scrollX Property
autoScroll Property
scrollTo Method
bringIntoView Method

DiagramView.scrollZoneSize Property

Gets or sets the size of the auto scroll zone near the edges of the control.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get scrollZoneSize() {}
set scrollZoneSize(value) {}
```

Property Value

Number

A number specifying the size of the auto scroll zone. The default value is 8.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
autoScroll Property

DiagramView.tooltipDelay Property

Gets or sets the delay before a tooltip is shown.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get tooltipDelay() {}
set tooltipDelay(value) {}
```

Property Value

Number

The tooltip delay in milliseconds.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.verticalDropAlignment Property

Gets or sets the vertical placement of nodes created by drag-and-drop operations, relatively to mouse cursor.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get verticalDropAlignment() {}
set verticalDropAlignment(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration. The default value is **Near**.

Remarks

This property lets you center the new node or align specified edge to the mouse pointer position when drag-and-drop ends.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
horizontalDropAlignment Property

DiagramView.viewport Property

Returns the visible portion of the diagram area.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
get viewport() {}
```

Property Value

Rect

A Rect instance specifying the viewport coordinates.

Remarks

When the view's canvas element is placed inside a scrollable div element, this method returns what part of the diagram is currently visible within the div. The result is specified in the current measure unit.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.virtualScroll Property

Gets or sets a value indicating whether virtual scroll mode is enabled.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
get virtualScroll() {}
set virtualScroll(value) {}
```

Property Value

Boolean

true if virtual scroll is enabled, or **false** otherwise. The default value is **true**.

Remarks

If this mode is enabled, the canvas stays as big as its containing div, and its content is drawn with a translation bound to the div's scrollbar positions. This allow displaying large diagrams in browsers and on mobile devices where the size of HTML canvas elements is limited.

Virtual scroll mode requires browser support for grid layout. If you need to use it in older browsers, set CompatConfig.gridLayout = false to use absolute positioning instead.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
CompatConfig.gridLayout Field

DiagramView.zoomFactor Property

Gets or sets the zoom factor.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
get zoomFactor() {}
set zoomFactor(value) {}
```

Property Value

Number

The current zoom factor as percentage.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView Events

DiagramView.createEditControl Event

Raised to let you create a custom inplace editor control.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
diagramView.createEditControl.addEventListener(onCreateEditControl);

function onCreateEditControl(sender, args)
{
}
```

Event Data

Event handlers receive an InplaceEditEventArgs instance as argument.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
allowInplaceEdit Property

DiagramView.diagramChanged Event

Raised when the diagram property changes.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
diagramView.diagramChanged.addEventListener(onDiagramChanged);

function onDiagramChanged(sender, args)
{
}
```

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DiagramView.enterInplaceEditMode Event

Raised when entering inplace-edit mode.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
diagramView.enterInplaceEditMode.addEventListener(onEnterInplaceEditMode);

function onEnterInplaceEditMode(sender, args)
{
}
```

Event Data

Event handlers receive an InplaceEditEventArgs instance as argument.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

allowInplaceEdit Property
leaveInplaceEditMode Event

DiagramView.leaveInplaceEditMode Event

Raised when leaving inplace-edit mode.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
diagramView.leaveInplaceEditMode.addEventListener(onLeaveInplaceEditMode);

function onLeaveInplaceEditMode(sender, args)
{
}
```

Event Data

Event handlers receive an InplaceEditEventArgs instance as argument.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
allowInplaceEdit Property
enterInplaceEditMode Event

DiagramView.mouseWheel Event

Raised when the user rotates the mouse wheel while the cursor is positioned over the diagram surface.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript Copy Code

```
diagramView.mouseWheel.addEventListener(onMouseWheel);

function onMouseWheel(sender, args)
{
}
```

Event Data

Event handlers receive a DiagramEventArgs instance as argument.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace
mouseWheelAction Property

DiagramView.zoomChanged Event

Raised when the zoom factor property changes.

Namespace: MindFusion.Diagramming
Assembly: DiagramView.js

Syntax

JavaScript — Copy Code

```
diagramView.zoomChanged.addEventListener(onZoomChanged);

function onZoomChanged(sender, args)
{
}
```

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming Namespace

DirectionConstraint Enumeration

Defines values that specify in what direction to constrain node movement.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
DirectionConstraint = {}
```

Members

	Member name	Description
	None	No restrictions.
	Horizontal	Users are allowed to move the node horizontally.
	Vertical	Users are allowed to move the node vertically.

See Also

MindFusion.Diagramming Namespace

DoNothingBehavior Class

Implements the DoNothing behavior.

Syntax

JavaScript — Copy Code

```
// class
DoNothingBehavior.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.BehaviorBase
  MindFusion.Diagramming.DoNothingBehavior

See Also

DoNothingBehavior Members
MindFusion.Diagramming Namespace

DoNothingBehavior Members

Implements the DoNothing behavior.

The following tables list the members exposed by the DoNothingBehavior type.

Public Constructors

	Name	Description
	DoNothingBehavior	Overloaded.

Public Properties

	Name	Description
	diagram	Gets the target diagram of this behavior object. (Inherited from BehaviorBase.)

Public Methods

	Name	Description
	createController	BehaviorBase.createController override.
	setMouseCursor	BehaviorBase.setMouseCursor override.

See Also

DoNothingBehavior Class
MindFusion.Diagramming Namespace

DoNothingBehavior Constructor

Overload List

Name	Description
DoNothingBehavior (Diagram)	Initializes a new instance of the DoNothingBehavior class.
DoNothingBehavior (DiagramView)	Initializes a new instance of the DoNothingBehavior class.

See Also

DoNothingBehavior Class
MindFusion.Diagramming Namespace

DoNothingBehavior Constructor (Diagram)

Initializes a new instance of the DoNothingBehavior class.

Namespace: MindFusion.Diagramming
Assembly: DoNothingBehavior.js

Syntax

JavaScript Copy Code

```
function DoNothingBehavior (parent)
```

Parameters

parent

The Diagram this behavior will be associated with.

See Also

DoNothingBehavior Constructor Overload List
DoNothingBehavior Class
MindFusion.Diagramming Namespace

DoNothingBehavior Constructor (DiagramView)

Initializes a new instance of the DoNothingBehavior class.

Namespace: MindFusion.Diagramming
Assembly: DoNothingBehavior.js

Syntax

JavaScript Copy Code

```
function DoNothingBehavior (parent)
```

Parameters

parent
DiagramView. The DiagramView this behavior will be associated with..

See Also

DoNothingBehavior Constructor Overload List
DoNothingBehavior Class
MindFusion.Diagramming Namespace

DoNothingBehavior Methods

DoNothingBehavior.createController Method

BehaviorBase.createController override.

Namespace: MindFusion.Diagramming
Assembly: DoNothingBehavior.js

Syntax

JavaScript Copy Code

```
function createController (state)
```

Parameters

state
Object. An object representing current context and input event.

Return Value

SinglePointerController. An instance of SinglePointerController -derived class.

See Also

DoNothingBehavior Members
DoNothingBehavior Class
MindFusion.Diagramming Namespace

DoNothingBehavior.setMouseCursor Method

BehaviorBase.setMouseCursor override.

Namespace: MindFusion.Diagramming
Assembly: DoNothingBehavior.js

Syntax

JavaScript Copy Code

```
function setMouseCursor (point, startInteraction)
```

Parameters

point

startInteraction

See Also

DoNothingBehavior Members
DoNothingBehavior Class
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs Class

Contains the arguments passed to the drawAnchorPoint event handlers.

Syntax

JavaScript Copy Code

```
// class
DrawAnchorPointEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.CancelEventArgs
  MindFusion.Diagramming.ItemEventArgs
    MindFusion.Diagramming.NodeEventArgs
      MindFusion.Diagramming.DrawAnchorPointEventArgs

See Also

DrawAnchorPointEventArgs Members
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs Members

Contains the arguments passed to the drawAnchorPoint event handlers.

The following tables list the members exposed by the DrawAnchorPointEventArgs type.

Public Constructors

	Name	Description
	DrawAnchorPointEventArgs	Initializes a new instance of the **DrawAnchorPointEventArgs** class.

Public Properties

	Name	Description
	AdjustmentHandle	Gets or sets the adjustment handle being moved by the user. (Inherited from NodeEventArgs.)
	anchorIndex	Gets the zero-based index of the anchor point related to the event.
	anchorPattern	Gets the anchor pattern containing the anchor point being drawn.

	Label	Gets or sets a NodeLabel related to this event. (Inherited from NodeEventArgs.)
	location	Gets the location of the anchor point being drawn.
	MouseButton	Gets which mouse button has been pressed. (Inherited from NodeEventArgs.)
	MousePosition	Gets or sets the current mouse pointer position. (Inherited from NodeEventArgs.)
	Node	Gets the node for which the event was raised. (Inherited from NodeEventArgs.)

See Also

DrawAnchorPointEventArgs Class
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs Constructor

Initializes a new instance of the DrawAnchorPointEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function DrawAnchorPointEventArgs (args)
```

Parameters

args

See Also

DrawAnchorPointEventArgs Class
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs Properties

DrawAnchorPointEventArgs.anchorIndex Property

Gets the zero-based index of the anchor point related to the event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get anchorIndex() {}
set anchorIndex(value) {}
```

Property Value

Number

The anchor point index.

See Also

DrawAnchorPointEventArgs Members
DrawAnchorPointEventArgs Class

DrawAnchorPointEventArgs.anchorPattern Property

Gets the anchor pattern containing the anchor point being drawn.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get anchorPattern() {}
set anchorPattern(value) {}
```

Property Value

AnchorPattern

The anchor pattern.

See Also

DrawAnchorPointEventArgs Members
DrawAnchorPointEventArgs Class
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs.location Property

Gets the location of the anchor point being drawn.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get location() {}
set location(value) {}
```

Property Value

Point

A Point instance.

See Also

DrawAnchorPointEventArgs Members
DrawAnchorPointEventArgs Class
MindFusion.Diagramming Namespace

DrawLinksBehavior Class

This class is used when the diagram behavior is set to Behavior.DrawLinks.

Syntax

JavaScript Copy Code

```
// class
DrawLinksBehavior.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.BehaviorBase
  MindFusion.Diagramming.DrawLinksBehavior

See Also

DrawLinksBehavior Members
MindFusion.Diagramming Namespace

DrawLinksBehavior Members

This class is used when the diagram behavior is set to Behavior.DrawLinks.

The following tables list the members exposed by the DrawLinksBehavior type.

Public Constructors

	Name	Description
	DrawLinksBehavior	Overloaded.

Public Properties

	Name	Description
	diagram	Gets the target diagram of this behavior object. (Inherited from BehaviorBase.)

Public Methods

	Name	Description
	createController	BehaviorBase.createController override.

See Also

DrawLinksBehavior Class
MindFusion.Diagramming Namespace

DrawLinksBehavior Constructor

Overload List

Name	Description
DrawLinksBehavior (Diagram)	Initializes a new instance of the DrawLinksBehavior class.
DrawLinksBehavior (DiagramView)	Initializes a new instance of the DrawLinksBehavior class.

See Also

DrawLinksBehavior Class
MindFusion.Diagramming Namespace

DrawLinksBehavior Constructor (Diagram)

Initializes a new instance of the DrawLinksBehavior class.

Namespace: MindFusion.Diagramming
Assembly: DrawLinksBehavior.js

Syntax

JavaScript Copy Code

```
function DrawLinksBehavior (parent)
```

Parameters

parent
The Diagram this behavior will be associated with.

See Also

DrawLinksBehavior Constructor Overload List
DrawLinksBehavior Class
MindFusion.Diagramming Namespace

DrawLinksBehavior Constructor (DiagramView)

Initializes a new instance of the DrawLinksBehavior class.

Namespace: MindFusion.Diagramming
Assembly: DrawLinksBehavior.js

Syntax

JavaScript Copy Code

```
function DrawLinksBehavior (parent)
```

Parameters

parent
DiagramView. The DiagramView this behavior will be associated with.

See Also

DrawLinksBehavior Constructor Overload List
DrawLinksBehavior Class
MindFusion.Diagramming Namespace

DrawLinksBehavior Methods

DrawLinksBehavior.createController Method

BehaviorBase.createController override.

Namespace: MindFusion.Diagramming
Assembly: DrawLinksBehavior.js

Syntax

JavaScript Copy Code

```
function createController (state)
```

Parameters

state
Object. An object representing current context and input event.

Return Value

SinglePointerController. An instance of SinglePointerController -derived class.

See Also

DrawLinksBehavior Members
DrawLinksBehavior Class
MindFusion.Diagramming Namespace

DrawNodesBehavior Class

The base class for behavior types that allow drawing of nodes, but not links.

Syntax

JavaScript Copy Code

```
// class
DrawNodesBehavior.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.BehaviorBase
  MindFusion.Diagramming.DrawNodesBehavior

See Also

DrawNodesBehavior Members
MindFusion.Diagramming Namespace

DrawNodesBehavior Members

The base class for behavior types that allow drawing of nodes, but not links.

The following tables list the members exposed by the DrawNodesBehavior type.

Public Constructors

	Name	Description
	DrawNodesBehavior	Overloaded.

Public Properties

	Name	Description
	diagram	Gets the target diagram of this behavior object. (Inherited from BehaviorBase.)

Public Methods

	Name	Description
	createController	BehaviorBase.createController override.

See Also

DrawNodesBehavior Class
MindFusion.Diagramming Namespace

DrawNodesBehavior Constructor

Overload List

Name	Description

DrawNodesBehavior (Diagram)	Initializes a new instance of the DrawNodesBehavior class.
DrawNodesBehavior (DiagramView)	Initializes a new instance of the DrawNodesBehavior class.

See Also

DrawNodesBehavior Class
MindFusion.Diagramming Namespace

DrawNodesBehavior Constructor (Diagram)

Initializes a new instance of the DrawNodesBehavior class.

Namespace: MindFusion.Diagramming
Assembly: DrawNodesBehavior.js

Syntax

JavaScript Copy Code

```
function DrawNodesBehavior (parent)
```

Parameters

parent
The Diagram this behavior will be associated with.

See Also

DrawNodesBehavior Constructor Overload List
DrawNodesBehavior Class
MindFusion.Diagramming Namespace

DrawNodesBehavior Constructor (DiagramView)

Initializes a new instance of the DrawNodesBehavior class.

Namespace: MindFusion.Diagramming
Assembly: DrawNodesBehavior.js

Syntax

JavaScript Copy Code

```
function DrawNodesBehavior (parent)
```

Parameters

parent
DiagramView. The DiagramView this behavior will be associated with.

See Also

DrawNodesBehavior Constructor Overload List
DrawNodesBehavior Class
MindFusion.Diagramming Namespace

DrawNodesBehavior Methods

DrawNodesBehavior.createController Method

BehaviorBase.createController override.

Namespace: MindFusion.Diagramming
Assembly: DrawNodesBehavior.js

Syntax

JavaScript — Copy Code

```
function createController (state)
```

Parameters

state
Object. An object representing current context and input event.

Return Value

SinglePointerController. An instance of SinglePointerController -derived class.

See Also

DrawNodesBehavior Members
DrawNodesBehavior Class
MindFusion.Diagramming Namespace

DrawShapesBehavior Class

This class is used when the diagram behavior is set to Behavior.DrawShapes.

Syntax

JavaScript — Copy Code

```
// class
DrawShapesBehavior.prototype = {}
```

Inheritance Hierarchy

See Also

DrawShapesBehavior Members
MindFusion.Diagramming Namespace

DrawShapesBehavior Members

This class is used when the diagram behavior is set to Behavior.DrawShapes.

The following tables list the members exposed by the DrawShapesBehavior type.

Public Constructors

	Name	Description
	DrawShapesBehavior	Initializes a new instance of the DrawShapesBehavior class.

Public Methods

	Name	Description
	createNode	DrawNodesBehavior.CreateNode override.

See Also

DrawShapesBehavior Class
MindFusion.Diagramming Namespace

DrawShapesBehavior Constructor

Initializes a new instance of the DrawShapesBehavior class.

Namespace: MindFusion.Diagramming
Assembly: DrawShapesBehavior.js

Syntax

JavaScript Copy Code

```
function DrawShapesBehavior (parent)
```

Parameters

parent
The Diagram this behavior will be associated with.

See Also

DrawShapesBehavior Class
MindFusion.Diagramming Namespace

DrawShapesBehavior Methods

DrawShapesBehavior.createNode Method

DrawNodesBehavior.CreateNode override.

Namespace: MindFusion.Diagramming
Assembly: DrawShapesBehavior.js

Syntax

JavaScript Copy Code

```
function createNode ()
```

Return Value

An instance of the DiagramNode class.

See Also

DrawShapesBehavior Members
DrawShapesBehavior Class
MindFusion.Diagramming Namespace

DummyNode Class

DummyNode represents virtual nodes set as targets of disconnected links.

Syntax

JavaScript Copy Code

```
// class
DummyNode.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.DummyNode

See Also

DummyNode Members
MindFusion.Diagramming Namespace

DummyNode Members

DummyNode represents virtual nodes set as targets of disconnected links.

The following tables list the members exposed by the **DummyNode** type.

Public Constructors

	Name	Description
	DummyNode	Initializes a new instance of the DummyNode class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)

	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)

	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)

	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)

	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object. (Inherited from DiagramNode.)
	LoadFromXml	DiagramItem.LoadFromXml override. (Inherited from DiagramNode.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)

	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from DiagramNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)

	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

DummyNode Class
MindFusion.Diagramming Namespace

DummyNode Constructor

Initializes a new instance of the DummyNode class.

Namespace: MindFusion.Diagramming
Assembly: DummyNode.js

Syntax

JavaScript Copy Code

```
function DummyNode ([parent])
```

Parameters

parent
Optional.
Diagram. A Diagram instance whose default node attributes are copied to this node.

See Also

DummyNode Class
MindFusion.Diagramming Namespace

EffectPhase Enumeration

Specifies the rendering phase during which an effect application is requested.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
EffectPhase = {}
```

Members

	Member name	Description
	BeforeFill	Indicates that the NodeEffect.apply method is invoked before the target node is filled.
	AfterFill	Indicates that the NodeEffect.apply method is invoked after the target node is filled.
	AfterOutline	Indicates that the NodeEffect.apply method is invoked after the target node is stroked.

See Also

MindFusion.Diagramming Namespace

Events Class

Obsolete. Defines all events raised by the Diagram component.

Syntax

JavaScript Copy Code

```
// class
Events.prototype = {}
```

Remarks

Obsolete

Events are now raised by event emitters exposed as properties of the Diagram class. Adding event listeners through this class is supported for compatibility with old code.

To register an event handler function, call the addEventListener method of **Diagram**. Event handlers receive the diagram as first argument, and an instance of EventArgs subclass as second argument.

Example

The following code handles nodeCreated event, raised when the user draws a new node.

JavaScript Copy Code

```
diagram.addEventListener(Events.nodeCreated, onNodeCreated);

function onNodeCreated(sender, args)
{
    var node = args.node;
    node.handlesStyle = HandlesStyle.HatchHandles3;
}
```

Inheritance Hierarchy

MindFusion.Diagramming.Events

See Also

Events Members
MindFusion.Diagramming Namespace
addEventListener Method
removeEventListener Method

Events Members

Obsolete. Defines all events raised by the Diagram component.

The following tables list the members exposed by the Events type.

Public Properties

	Name	Description
	actionRedone	Raised when an action is redone.
	actionUndone	Raised when an action is undone.
	animatedLayoutCompleted	Raised when an animated layout completes.
	cellTextEdited	Raised when the user edits the text of a table cell.

	cellTextEditing	Raised when the user starts to edit the text of a table cell.
	clicked	Raised when the user clicks the diagram at a point where no items are located.
	containerChildAdded	Raised when the user adds a node to a ContainerNode.
	containerChildAdding	A validation event raised to let you prevent users from adding child nodes to a container.
	containerChildRemoved	Raised when the user removes a node from a **ContainerNode**.
	containerChildRemoving	A validation event raised to let you prevent users from removing child nodes from a container.
	containerFolded	Raised when the user folds a container by clicking the arrow-up button in the container's caption bar.
	containerPageChanged	Raised when the user changes current page of a PagedContainerNode.
	containerUnfolded	Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar.
	controlLoaded	Raised when the control is loaded. For internal use by MindFusion ASP.NET WebForms / MVC controls.
	controlUnloaded	Raised when the control is unloaded.
	controlUnloading	Raised before the control is unloaded.
	createEditControl	Raised to let you create a custom inplace editor control.
	dataLoaded	Raised when the diagram data is loaded.
	deserializeTag	Raised when the Tag or Id of a diagram item must be deserialized from XML format.
	diagramChanged	Raised when the diagram property of a diagram view changes.
	documentChanged	Raised when a page is added or removed from a DiagramDocument instance.
	drawAdjustmentHandles	Raised to draw custom adjustment handles.
	drawAnchorPoint	Raised when an anchor point must be custom-drawn.
	enterInplaceEditMode	Raised when entering inplace-edit mode.
	expandButtonClicked	Raised if ExpandButtonAction is set to RaiseEvents and the [+/-]button of an Expandable node is clicked.
	headerResized	Raised when the user finished resizing a header.
	headerStartResizing	Raised when the user starts resizing a header.
	hitTestAdjustmentHandles	Raised to hit-test custom adjustment handles.
	hyperlinkClicked	Raised when the user clicks a hyperlink inside a node.
	initializeLink	Raised when the user starts drawing a new link, just after the DiagramLink instance is created.
	initializeNode	Raised when the user starts drawing a new node, just after the DiagramNode instance is created.
	itemAdded	Raised when an item is added to the diagram.
	itemRemoved	Raised when an item is removed from the diagram.
	laneGridCellTextEdited	Raised when the user edits the text of a grid cell.
	leaveInplaceEditMode	Raised when leaving inplace-edit mode.
	linkClicked	Raised when the user clicks a link.

	linkCreated	Raised when the user draws a new link.
	linkCreating	A validation event raised while the user is drawing a new link.
	linkDeleted	Raised when a link is deleted, either programmatically or by the user.
	linkDeleting	Raised when the user tries to delete a link. This event lets you cancel the operation.
	linkDeselected	Raised when a link is deselected, either programmatically or by the user.
	linkDoubleClicked	Raised when the user double-clicks a link.
	linkHovered	Raised when the mouse pointer hovers over a link.
	linkModified	Raised when the user moves any control point of a link.
	linkModifying	A validation event raised while the user is modifying a link.
	linkPasted	Raised when a link has been pasted from the clipboard.
	linkPointerEnter	Raised when the mouse pointer enters the boundaries of a link.
	linkPointerLeave	Raised when the mouse pointer leaves the boundaries of a link.
	linkRouted	Raised when a link is routed by the auto-routing algorithm. Event handlers receive a LinkEventArgs instance that contains data about this event.
	linkSelected	Raised when a link is selected, either programmatically or by the user.
	linkSelecting	Raised when the user tries to select a link. This event lets you cancel the operation.
	linkSplit	Raised when the user splits a link segment interactively.
	linkTextEdited	Raised when the user edits the text of a link.
	linkTextEditing	Raised when the user starts to edit the text of a link.
	mouseWheel	Raised when the user rotates the mouse wheel while the cursor is positioned over the diagram surface.
	nodeClicked	Raised when the user clicks a node.
	nodeCreated	Raised when the user draws a new node.
	nodeCreating	A validation event raised while the user is drawing a new node.
	nodeDeleted	Raised when a node is deleted, either programmatically or by the user.
	nodeDeleting	Raised when the user tries to delete a node. This event lets you cancel the operation.
	nodeDeselected	Raised when a node is deselected, either programmatically or by the user.
	nodeDomCreated	Raised when the content of a ControlNode is added to the page DOM.
	nodeDoubleClicked	Raised when the user double-clicks a node.
	nodeHovered	Raised when the mouse pointer hovers over a node.
	nodeModified	Raised when the user moves or resizes a node.
	nodeModifying	A validation event raised while the user is moving or resizing a node.
	nodePasted	Raised when a node has been pasted from the clipboard.
	nodePointerEnter	Raised when the mouse pointer enters the boundaries of a node.
	nodePointerLeave	Raised when the mouse pointer leaves the boundaries of a node.
	nodeSelected	Raised when a node is selected, either programmatically or by the user.

	nodeSelecting	Raised when the user tries to select a node. This event lets you cancel the operation.
	nodeTextEdited	Raised when the user edits the text of a node.
	nodeTextEditing	Raised when the user starts to edit the text of a node.
	queryView	Raised to enumerate diagram views.
	repaint	Raised when the diagram has been repainted.
	selectionChanged	Raised when the diagram selection has changed.
	selectionModifying	A validation event raised while the user is moving or resizing multiple items.
	serializeTag	Raised when the tag or id of a diagram item must be serialized in XML format.
	shapeLibraryLoaded	Raised when a ShapeLibrary is loaded from shapeLibraryLocation.
	sizeChanged	Raised when the size of the diagram canvas has changed.
	tableColumnResized	Raised when the user finishes resizing a column in a table.
	tableColumnResizing	Raised when the user resizes a column in a table interactively.
	tableRowResized	Raised when the user finishes resizing a row in a table.
	tableRowResizing	Raised when the user resizes a row in a table interactively.
	tableSectionCollapsed	Raised when a table section is collapsed.
	tableSectionExpanded	Raised when a table section is expanded.
	treeCollapsed	Raised when a user collapses a tree branch by clicking the [-] button of an expandable node.
	treeExpanded	Raised when a user expands a tree branch by clicking the [+] button of an **expandable** node.
	treeItemTextEdited	Raised when the user edits the text of a tree view item.
	treeItemTextEditing	Raised when the user starts to edit the text of a tree view item.
	validateAnchorPoint	A validation event that lets you approve or reject connecting a link to an anchor point.
	validateLinkCrossing	A validation event that lets you approve or reject displaying a decoration at the intersection point of links.

See Also

Events Class
MindFusion.Diagramming Namespace

Events Properties

Events.actionRedone Property

Raised when an action is redone.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get actionRedone() {}
set actionRedone(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.actionUndone Property

Raised when an action is undone.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get actionUndone() {}
set actionUndone(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.animatedLayoutCompleted Property

Raised when an animated layout completes.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get animatedLayoutCompleted() {}
set animatedLayoutCompleted(value) {}
```

Property Value

String

A string containing the event name.

Remarks

When layout is applied on a ContainerNode, event handlers receive a NodeEventArgs instance as argument.

Example

This example handles the **animatedLayoutCompleted** event to resize the diagram.

JavaScript Copy Code

```
diagram.addEventListener(Events.animatedLayoutCompleted, (sender, args) => {
    sender.resizeToFitItems(10);
});
diagram.arrangeAnimated(...);
```

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.cellTextEdited Property

Raised when the user edits the text of a table cell.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get cellTextEdited() {}
set cellTextEdited(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a CellEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.cellTextEditing Property

Raised when the user starts to edit the text of a table cell.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get cellTextEditing() {}
set cellTextEditing(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a CellEventArgs instance as argument.

See Also

Events Members
Events Class

MindFusion.Diagramming Namespace

Events.clicked Property

Raised when the user clicks the diagram at a point where no items are located.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get clicked() {}
set clicked(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a DiagramEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
linkClicked Event
nodeClicked Event

Events.containerChildAdded Property

Raised when the user adds a node to a ContainerNode.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get containerChildAdded() {}
set containerChildAdded(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.containerChildAdding Property

A validation event raised to let you prevent users from adding child nodes to a container.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get containerChildAdding() {}
set containerChildAdding(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.containerChildRemoved Property

Raised when the user removes a node from a ContainerNode.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get containerChildRemoved() {}
set containerChildRemoved(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.containerChildRemoving Property

A validation event raised to let you prevent users from removing child nodes from a container.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get containerChildRemoving() {}
set containerChildRemoving(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.containerFolded Property

Raised when the user folds a container by clicking the arrow-up button in the container's caption bar.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get containerFolded() {}
set containerFolded(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.containerPageChanged Property

Raised when the user changes current page of a PagedContainerNode.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get containerPageChanged() {}
set containerPageChanged(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events.containerUnfolded Property

Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript — Copy Code

```
get containerUnfolded() {}
set containerUnfolded(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.controlLoaded Property

Raised when the control is loaded. For internal use by MindFusion ASP.NET WebForms / MVC controls.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript — Copy Code

```
get controlLoaded() {}
set controlLoaded(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Unlike most events listed here, the event listener must be added to DiagramView instead of Diagram, and respectively the *sender* argument of handler function is set to the view.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
addEventListener Method

Events.controlUnloaded Property

Raised when the control is unloaded.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get controlUnloaded() {}
set controlUnloaded(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.controlUnloading Property

Raised before the control is unloaded.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get controlUnloading() {}
set controlUnloading(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.createEditControl Property

Raised to let you create a custom inplace editor control.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get createEditControl() {}
set createEditControl(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive an InplaceEditEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.dataLoaded Property

Raised when the diagram data is loaded.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get dataLoaded() {}
set dataLoaded(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.deserializeTag Property

Raised when the Tag or Id of a diagram item must be deserialized from XML format.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get deserializeTag() {}
set deserializeTag(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a SerializeTagEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.diagramChanged Property

Raised when the diagram property of a diagram view changes.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get diagramChanged() {}
set diagramChanged(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.documentChanged Property

Raised when a page is added or removed from a DiagramDocument instance.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get documentChanged() {}
set documentChanged(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.drawAdjustmentHandles Property

Raised to draw custom adjustment handles.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get drawAdjustmentHandles() {}
set drawAdjustmentHandles(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive either NodeEventArgs or LinkEventArgs instance as argument, depending on the item type.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.drawAnchorPoint Property

Raised when an anchor point must be custom-drawn.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get drawAnchorPoint() {}
set drawAnchorPoint(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Set the markStyle property of AnchorPoint to Custom to get this event raised. Handlers receive a DrawAnchorPointEventArgs instance as argument.

Example

JavaScript Copy Code

```
diagram.addEventListener(
    Events.drawAnchorPoint, onDrawAnchorPoint);

function onDrawAnchorPoint(sender, e)
{
    var p = e.location;
    var ctx = e.context;
    ctx.fillStyle = "red";
    ctx.fillRect(p.x - 1, p.y - 1, 2, 2);
}
```

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.enterInplaceEditMode Property

Raised when entering inplace-edit mode.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get enterInplaceEditMode() {}
set enterInplaceEditMode(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive an InplaceEditEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.expandButtonClicked Property

Raised if ExpandButtonAction is set to RaiseEvents and the [+/-]button of an Expandable node is clicked.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get expandButtonClicked() {}
set expandButtonClicked(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.headerResized Property

Raised when the user finished resizing a header.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get headerResized() {}
set headerResized(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a HeaderResizeEventArgs instance as argument.

See Also

Events.headerStartResizing Property

Raised when the user starts resizing a header.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript — Copy Code

```
get headerStartResizing() {}
set headerStartResizing(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a HeaderResizeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.hitTestAdjustmentHandles Property

Raised to hit-test custom adjustment handles.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript — Copy Code

```
get hitTestAdjustmentHandles() {}
set hitTestAdjustmentHandles(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive either NodeEventArgs or LinkEventArgs instance as argument, depending on the item type.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.hyperlinkClicked Property

Raised when the user clicks a hyperlink inside a node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get hyperlinkClicked() {}
set hyperlinkClicked(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.initializeLink Property

Raised when the user starts drawing a new link, just after the DiagramLink instance is created.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get initializeLink() {}
set initializeLink(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.initializeNode Property

Raised when the user starts drawing a new node, just after the DiagramNode instance is created.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get initializeNode() {}
set initializeNode(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.itemAdded Property

Raised when an item is added to the diagram.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get itemAdded() {}
set itemAdded(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive an ItemEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.itemRemoved Property

Raised when an item is removed from the diagram.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get itemRemoved() {}
set itemRemoved(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive an ItemEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.laneGridCellTextEdited Property

Raised when the user edits the text of a grid cell.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get laneGridCellTextEdited() {}
set laneGridCellTextEdited(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a CellEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.leaveInplaceEditMode Property

Raised when leaving inplace-edit mode.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get leaveInplaceEditMode() {}
set leaveInplaceEditMode(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive an InplaceEditEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkClicked Property

Raised when the user clicks a link.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkClicked() {}
set linkClicked(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
linkHovered Event
linkDoubleClicked Event

Events.linkCreated Property

Raised when the user draws a new link.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkCreated() {}
set linkCreated(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkCreating Property

A validation event raised while the user is drawing a new link.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkCreating() {}
set linkCreating(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

Example

This example shows how to handle the **linkCreating** event to prevent cycles in the graph.

JavaScript Copy Code

```
// assuming namespaced scripts
var Events = MindFusion.Diagramming.Events;
var PathFinder = MindFusion.Diagramming.PathFinder;

diagram.addEventListener(
    Events.linkCreating, onLinkCreating);

function onLinkCreating(diagram, args)
{
    if (args.destination == null)
    {
        // not pointing to a node yet
        return;
    }

    var pathFinder = new PathFinder(diagram);
    var path = pathFinder.findShortestPath(
        args.destination, args.origin);

    if (path != null)
    {
        // adding this new link would create a cycle
        // [origin]--[dest]--[path internal nodes]--[origin]

        args.cancel = true;
    }
}
```

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
allowSelfLoops Property
allowUnconnectedLinks Property

Events.linkDeleted Property

Raised when a link is deleted, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkDeleted() {}
set linkDeleted(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkDeleting Property

Raised when the user tries to delete a link. This event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkDeleting() {}
set linkDeleting(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkDeselected Property

Raised when a link is deselected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkDeselected() {}
set linkDeselected(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkDoubleClicked Property

Raised when the user double-clicks a link.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkDoubleClicked() {}
set linkDoubleClicked(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkHovered Property

Raised when the mouse pointer hovers over a link.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkHovered() {}
set linkHovered(value) {}
```

Property Value

String

A string containing the event name.

Remarks

This event fires when the mouse pointer rests over a link for hoverDelay milliseconds. Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
linkClicked Event

linkPointerEnter Event
linkPointerLeave Event

Events.linkModified Property

Raised when the user moves any control point of a link.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkModified() {}
set linkModified(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkModifying Property

A validation event raised while the user is modifying a link.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkModifying() {}
set linkModifying(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkPasted Property

Raised when a link has been pasted from the clipboard.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkPasted() {}
set linkPasted(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkPointerEnter Property

Raised when the mouse pointer enters the boundaries of a link.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkPointerEnter() {}
set linkPointerEnter(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance that contains data about this event.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
linkHovered Event
linkPointerLeave Event

Events.linkPointerLeave Property

Raised when the mouse pointer leaves the boundaries of a link.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkPointerLeave() {}
set linkPointerLeave(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance that contains data about this event.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
linkHovered Event
linkPointerEnter Event

Events.linkRouted Property

Raised when a link is routed by the auto-routing algorithm. Event handlers receive a LinkEventArgs instance that contains data about this event.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkRouted() {}
set linkRouted(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkSelected Property

Raised when a link is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkSelected() {}
set linkSelected(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkSelecting Property

Raised when the user tries to select a link. This event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkSelecting() {}
set linkSelecting(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkSplit Property

Raised when the user splits a link segment interactively.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkSplit() {}
set linkSplit(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
allowSplitLinks Property

Events.linkTextEdited Property

Raised when the user edits the text of a link.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkTextEdited() {}
set linkTextEdited(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.linkTextEditing Property

Raised when the user starts to edit the text of a link.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get linkTextEditing() {}
set linkTextEditing(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a LinkEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.mouseWheel Property

Raised when the user rotates the mouse wheel while the cursor is positioned over the diagram surface.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get mouseWheel() {}
set mouseWheel(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a DiagramEventArgs instance as argument. Unlike most events listed here, the event listener must be added to DiagramView instead of Diagram, and respectively the *sender* argument of handler function is set to the view.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeClicked Property

Raised when the user clicks a node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeClicked() {}
set nodeClicked(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
nodeDoubleClicked Event
nodeHovered Event

Events.nodeCreated Property

Raised when the user draws a new node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeCreated() {}
set nodeCreated(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

Example

This example handles the **nodeCreated** event to set new node's anchorPattern.

JavaScript Copy Code

```
diagram.addEventListener(Events.nodeCreated, (sender, args) => {
    if (sender.defaultShape.id == "Rectangle")
        args.node.anchorPattern = pattern1;
    else
        args.node.anchorPattern = pattern2;
});
```

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeCreating Property

A validation event raised while the user is drawing a new node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeCreating() {}
set nodeCreating(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeDeleted Property

Raised when a node is deleted, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeDeleted() {}
set nodeDeleted(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeDeleting Property

Raised when the user tries to delete a node. This event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeDeleting() {}
set nodeDeleting(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeDeselected Property

Raised when a node is deselected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeDeselected() {}
set nodeDeselected(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeDomCreated Property

Raised when the content of a ControlNode is added to the page DOM.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeDomCreated() {}
set nodeDomCreated(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeDoubleClicked Property

Raised when the user double-clicks a node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeDoubleClicked() {}
set nodeDoubleClicked(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeHovered Property

Raised when the mouse pointer hovers over a node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeHovered() {}
set nodeHovered(value) {}
```

Property Value

String

A string containing the event name.

Remarks

This event fires when the mouse pointer rests over a node for hoverDelay milliseconds. Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
nodeClicked Event
nodePointerEnter Event
nodePointerLeave Event

Events.nodeModified Property

Raised when the user moves or resizes a node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeModified() {}
set nodeModified(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeModifying Property

A validation event raised while the user is moving or resizing a node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeModifying() {}
set nodeModifying(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodePasted Property

Raised when a node has been pasted from the clipboard.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodePasted() {}
set nodePasted(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodePointerEnter Property

Raised when the mouse pointer enters the boundaries of a node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodePointerEnter() {}
set nodePointerEnter(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance that contains data about this event.

See Also

Events.nodePointerLeave Property

Raised when the mouse pointer leaves the boundaries of a node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodePointerLeave() {}
set nodePointerLeave(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance that contains data about this event.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
nodeHovered Event
nodePointerEnter Event

Events.nodeSelected Property

Raised when a node is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeSelected() {}
set nodeSelected(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeSelecting Property

Raised when the user tries to select a node. This event lets you cancel the operation.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript — Copy Code

```
get nodeSelecting() {}
set nodeSelecting(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeTextEdited Property

Raised when the user edits the text of a node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript — Copy Code

```
get nodeTextEdited() {}
set nodeTextEdited(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.nodeTextEditing Property

Raised when the user starts to edit the text of a node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get nodeTextEditing() {}
set nodeTextEditing(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.queryView Property

Raised to enumerate diagram views.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get queryView() {}
set queryView(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.repaint Property

Raised when the diagram has been repainted.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get repaint() {}
set repaint(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive an EventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.selectionChanged Property

Raised when the diagram selection has changed.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get selectionChanged() {}
set selectionChanged(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a SelectionChangedEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.selectionModifying Property

A validation event raised while the user is moving or resizing multiple items.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get selectionModifying() {}
set selectionModifying(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a SelectionEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.serializeTag Property

Raised when the tag or id of a diagram item must be serialized in XML format.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get serializeTag() {}
set serializeTag(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a SerializeTagEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.shapeLibraryLoaded Property

Raised when a ShapeLibrary is loaded from shapeLibraryLocation.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get shapeLibraryLoaded() {}
set shapeLibraryLoaded(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.sizeChanged Property

Raised when the size of the diagram canvas has changed.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get sizeChanged() {}
set sizeChanged(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive an EventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.tableColumnResized Property

Raised when the user finishes resizing a column in a table.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get tableColumnResized() {}
set tableColumnResized(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a CellEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
allowResizeColumns Property

Events.tableColumnResizing Property

Raised when the user resizes a column in a table interactively.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get tableColumnResizing() {}
set tableColumnResizing(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a CellEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
allowResizeColumns Property

Events.tableRowResized Property

Raised when the user finishes resizing a row in a table.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get tableRowResized() {}
set tableRowResized(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a CellEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
allowResizeRows Property

Events.tableRowResizing Property

Raised when the user resizes a row in a table interactively.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get tableRowResizing() {}
set tableRowResizing(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a CellEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace
allowResizeRows Property

Events.tableSectionCollapsed Property

Raised when a table section is collapsed.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get tableSectionCollapsed() {}
set tableSectionCollapsed(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.tableSectionExpanded Property

Raised when a table section is expanded.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get tableSectionExpanded() {}
set tableSectionExpanded(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.treeCollapsed Property

Raised when a user collapses a tree branch by clicking the [-] button of an expandable node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get treeCollapsed() {}
set treeCollapsed(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.treeExpanded Property

Raised when a user expands a tree branch by clicking the [+] button of an expandable node.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get treeExpanded() {}
set treeExpanded(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a NodeEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.treeItemTextEdited Property

Raised when the user edits the text of a tree view item.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get treeItemTextEdited() {}
set treeItemTextEdited(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a TreeItemEventArgs instance as argument.

Example

This example handles the **treeItemTextEdited** event to remove items without text.

JavaScript Copy Code

```
diagram.addEventListener(Events.treeItemTextEdited, (sender, args) => {
    if (!args.newText)
        args.node.removeItem(args.item);
});
```

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.treeItemTextEditing Property

Raised when the user starts to edit the text of a tree view item.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get treeItemTextEditing() {}
set treeItemTextEditing(value) {}
```

Property Value

String

A string containing the event name.

Remarks

Event handlers receive a TreeItemEventArgs instance as argument.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.validateAnchorPoint Property

A validation event that lets you approve or reject connecting a link to an anchor point.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get validateAnchorPoint() {}
set validateAnchorPoint(value) {}
```

Property Value

String

A string containing the event name.

Remarks

validateAnchorPoint event handlers receive an argument of type LinkValidationEventArgs.

This event is raised for each point of an AnchorPattern while a link's end point is being moved over a node. If none of the anchor points are validated, the mouse cursor icon changes to *NotAllowed*, indicating that the link cannot be connected to that node.

When the mouse button is released over a node, the event is raised again for the node's anchor points. From all validated anchor points, the one nearest to the dragged link end point is used to dock the link.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

Events.validateLinkCrossing Property

A validation event that lets you approve or reject displaying a decoration at the intersection point of links.

Namespace: MindFusion.Diagramming
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get validateLinkCrossing() {}
set validateLinkCrossing(value) {}
```

Property Value

String

A string containing the event name.

Remarks

validateLinkCrossing event handlers receive an argument of type LinkCrossingEventArgs.

This event is raised for each intersection point of links when linkCrossings property is enabled.

See Also

Events Members
Events Class
MindFusion.Diagramming Namespace

ExpandButtonAction Enumeration

Specifies what action is triggered by the +/- buttons that are displayed near expandable nodes.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
ExpandButtonAction = {}
```

Members

	Member name	Description
	ExpandTreeBranch	Expands or collapses the children of a node in a tree-like diagram and raises the treeExpanded or treeCollapsed event.
	RaiseEvents	Raises the expandButtonClicked event.

Remarks

Members of this enumeration can be assigned to the expandButtonAction property of Diagram.

See Also

MindFusion.Diagramming Namespace

Factory Class

Provides shortcut methods for creating and adding instances of the standard item types.

Syntax

JavaScript Copy Code

```
// class
Factory.prototype = {}
```

Remarks

Obtain an instance of this class through the factory property of the Diagram class.

Inheritance Hierarchy

MindFusion.Diagramming.Factory

See Also

Factory Members
MindFusion.Diagramming Namespace

Factory Members

Provides shortcut methods for creating and adding instances of the standard item types.

The following tables list the members exposed by the Factory type.

Public Constructors

	Name	Description
	Factory	Initializes a new instance of the Factory class over the specified Diagram.

Public Methods

	Name	Description
	createContainerNode	Creates a new ContainerNode instance and adds it to the nodes collection of the underlying diagram.
	createControlNode	Creates a new ControlNode instance and adds it to the nodes collection of the underlying diagram.
	createDiagramLink	Overloaded.
	createPagedContainerNode	Creates a new PagedContainerNode instance and adds it to the nodes collection of the underlying diagram.
	createShapeNode	Creates a new ShapeNode instance and adds it to the nodes collection of the underlying diagram.
	createSvgNode	Creates a new SvgNode instance and adds it to the nodes collection of the underlying diagram.
	createTableNode	Creates a new TableNode instance and adds it to the nodes collection of the underlying diagram.
	createTreeViewNode	Creates a new TreeViewNode instance and adds it to the nodes collection of the underlying diagram.

	createVideoNode	Creates a new VideoNode instance and adds it to the nodes collection of the underlying diagram.

See Also

Factory Class
MindFusion.Diagramming Namespace

Factory Constructor

Initializes a new instance of the Factory class over the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: Factory.js

Syntax

JavaScript Copy Code

```
function Factory (diagram)
```

Parameters

diagram
Diagram. The Diagram object for which to create the new factory.

See Also

Factory Class
MindFusion.Diagramming Namespace

Factory Methods

Factory.createContainerNode Method

Creates a new ContainerNode instance and adds it to the nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: Factory.js

Syntax

JavaScript Copy Code

```
function createContainerNode ([bounds, [x, [y, [width, [height]]]]])
```

Parameters

bounds
Optional. A Rect instance specifying the bounding rectangle of the new node.

x
Optional.
Number. The x-coordinate of the upper-left corner of the new node.

y
Optional.
Number. The y-coordinate of the upper-left corner of the new node.

width
Optional.
Number. The width of the new node.

height
Optional.
Number. The height of the new node.

Return Value

The newly created ContainerNode instance.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.createControlNode Method

Creates a new ControlNode instance and adds it to the nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: Factory.js

Syntax

JavaScript Copy Code

```
function createControlNode ([bounds, [x, [y, [width, [height]]]]])
```

Parameters

bounds
Optional.
Rect. Specifies the bounding rectangle of the new node.

x
Optional.
Number. The x-coordinate of the upper-left corner of the new node.

y
Optional.
Number. The y-coordinate of the upper-left corner of the new node.

width
Optional.
Number. The width of the new node.

height
Optional.
Number. The height of the new node.

Return Value

ControlNode. The newly created ControlNode instance.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.createDiagramLink Method

Overload List

Name	Description
Factory.createDiagramLink (DiagramNode, DiagramNode)	Creates a new DiagramLink instance between the specified nodes and adds it to the links collection of the underlying diagram.
Factory.createDiagramLink (DiagramNode, Number | TreeViewItem, DiagramNode, Number | TreeviewItem)	Creates a new **DiagramLink** instance between the specified nodes and adds it to the links collection of the underlying diagram.

See Also

Factory Members
Factory Class

Factory.createDiagramLink Method (DiagramNode, DiagramNode)

Creates a new DiagramLink instance between the specified nodes and adds it to the links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: Factory.js

Syntax

JavaScript Copy Code

```
function createDiagramLink ([origin, [destination]])
```

Parameters

origin
Optional. The origin node.

destination
Optional. The destination **node**.

Return Value

The newly created DiagramLink instance.

Remarks

The method creates a link connecting the nodes specified by origin and destination.

See Also

createDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.createDiagramLink Method (DiagramNode, Number | TreeViewItem, DiagramNode, Number | TreeviewItem)

Creates a new DiagramLink instance between the specified nodes and adds it to the links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: Factory.js

Syntax

JavaScript Copy Code

```
function createDiagramLink (
    origin, [originRow], destination, [destRow])
```

Parameters

origin
Optional.
DiagramNode. The origin node.

originRow
Optional.
Number | TreeViewItem. The zero-based index of the origin table row or the origin tree item of the new link.

destination
Optional.
DiagramNode. The destination node.

destRow
Optional.
Number | **TreeviewItem**. The zero-based index of the destination table row or the destination tree item of the new link.

Return Value

The newly created DiagramLink instance.

Remarks

If the *origin* or *destination* argument refers to a TableNode, it can be optionally followed by a row index specifying the origin or destination row. If the *origin* or *destination* argument refers to a TreeViewNode, it can be optionally followed by a TreeViewItem argument specifying the origin or destination item. If index / item arguments are omitted, the link connects to the specified node as a target instead of specific element inside.

Example

The following script demonstrates various Factory method calls.

JavaScript Copy Code

```
function buildDiagram(diagram)
{
    var node = diagram.factory.createShapeNode(10, 50, 40, 30);

    var table = diagram.factory.createTableNode(80, 10, 40, 60);
    table.text = "table";
    table.rowCount = 10;

    var treeView = diagram.factory.createTreeViewNode(150, 90, 40, 60);
    treeView.addItem(new TreeViewItem());
    treeView.addItem(new TreeViewItem());

    diagram.factory.createDiagramLink(node, table, 2);
    diagram.factory.createDiagramLink(table, 4, node);

    diagram.factory.createDiagramLink(treeView, treeView.rootItems[1], node);

    diagram.factory.createDiagramLink(table, 6, treeView, treeView.rootItems[0]);
}
```

See Also

createDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.createPagedContainerNode Method

Creates a new PagedContainerNode instance and adds it to the nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: Factory.js

Syntax

JavaScript Copy Code

```
function createPagedContainerNode ([bounds, [x, [y, [width, [height, [pageTitles]]]]]])
```

Parameters

bounds
Optional.
Rect. Specifies the bounding rectangle of the new node.

x
Optional.
Number. The x-coordinate of the upper-left corner of the new node.

y
Optional.
Number. The y-coordinate of the upper-left corner of the new node.

width
Optional.

Number. The width of the new node.

height
Optional.
Number. The height of the new node.

pageTitles
Optional.
Array. An array of title strings for the initial pages.

Return Value

PagedContainerNode. The newly created PagedContainerNode instance.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.createShapeNode Method

Creates a new ShapeNode instance and adds it to the nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: Factory.js

Syntax

JavaScript Copy Code

```
function createShapeNode ([bounds, [x, [y, [width, [height]]]]])
```

Parameters

bounds
Optional. A Rect instance specifying the bounding rectangle of the new node.

x
Optional.
Number. The x-coordinate of the upper-left corner of the new node.

y
Optional.
Number. The y-coordinate of the upper-left corner of the new node.

width
Optional.
Number. The width of the new node.

height
Optional.
Number. The height of the new node.

Return Value

The newly created ShapeNode instance.

Remarks

This method accepts either a single Rect object specifying the bounding rectangle of the new node, or a list of four numbers specifying the location and size.

The method creates a shape node with the specified arguments, if any.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.createSvgNode Method

Creates a new SvgNode instance and adds it to the nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: Factory.js

Syntax

JavaScript Copy Code

```
function createSvgNode ([bounds, [x, [y, [width, [height]]]]])
```

Parameters

bounds
Optional. A Rect instance specifying the bounding rectangle of the new node.

x
Optional.
Number. The x-coordinate of the upper-left corner of the new node.

y
Optional.
Number. The y-coordinate of the upper-left corner of the new node.

width
Optional.
Number. The width of the new node.

height
Optional.
Number. The height of the new node.

Return Value

The newly created SvgNode instance.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.createTableNode Method

Creates a new TableNode instance and adds it to the nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: Factory.js

Syntax

JavaScript Copy Code

```
function createTableNode ([bounds, [x, [y, [width, [height]]]]])
```

Parameters

bounds
Optional. A Rect instance specifying the bounding rectangle of the new node.

x
Optional.
Number. The x-coordinate of the upper-left corner of the new node.

y
Optional.
Number. The y-coordinate of the upper-left corner of the new node.

width
Optional.
Number. The width of the new node.

height
Optional.
Number. The height of the new node.

Return Value

The newly created TableNode instance.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.createTreeViewNode Method

Creates a new TreeViewNode instance and adds it to the nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: Factory.js

Syntax

JavaScript Copy Code

```
function createTreeViewNode ([bounds, [x, [y, [width, [height]]]]])
```

Parameters

bounds
Optional.
Rect. Specifies the bounding rectangle of the new node.

x
Optional.
Number. The x-coordinate of the upper-left corner of the new node.

y
Optional.
Number. The y-coordinate of the upper-left corner of the new node.

width
Optional.
Number. The width of the new node.

height
Optional.
Number. The height of the new node.

Return Value

TreeViewNode. The newly created TreeViewNode instance.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.createVideoNode Method

Creates a new VideoNode instance and adds it to the nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: Factory.js

Syntax

JavaScript Copy Code

```
function createVideoNode ([bounds, [x, [y, [width, [height]]]]])
```

Parameters

bounds
Optional.
A Rect instance specifying the bounding rectangle of the new node.

x
Optional.

Number. The x-coordinate of the upper-left corner of the new node.

y
Optional.
Number. The y-coordinate of the upper-left corner of the new node.

width
Optional.
Number. The width of the new node.

height
Optional.
Number. The height of the new node.

Return Value

The newly created VideoNode instance.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

FitSize Enumeration

Specifies how the resizeToFitText method should resize a node.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
FitSize = {}
```

Members

	Member name	Description
	KeepWidth	The original node width should remain intact.
	KeepHeight	The original node height should remain intact.
	KeepRatio	Allow changing both the width and height of a node, but keep the original width/height ratio if possible.

See Also

MindFusion.Diagramming Namespace

FoldContainerCommand Class

Handles undo / redo of container folding.

Syntax

JavaScript Copy Code

```
// class
FoldContainerCommand.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Command
  MindFusion.Diagramming.FoldContainerCommand

See Also

FoldContainerCommand Members
MindFusion.Diagramming Namespace

FoldContainerCommand Members

Handles undo / redo of container folding.

The following tables list the members exposed by the FoldContainerCommand type.

Public Constructors

	Name	Description
	FoldContainerCommand	Initializes a new instance of the **FoldContainerCommand** class.

Public Methods

	Name	Description
	execute	Command.execute override.
	redo	Command.redo override.
	undo	Command.undo override.

See Also

FoldContainerCommand Class
MindFusion.Diagramming Namespace

FoldContainerCommand Constructor

Initializes a new instance of the FoldContainerCommand class.

Namespace: MindFusion.Diagramming
Assembly: FoldContainerCommand.js

Syntax

JavaScript Copy Code

```
function FoldContainerCommand (diagram, container)
```

Parameters

diagram
The Diagram modified by this command.

container
The ContainerNode instance that will be modified.

See Also

FoldContainerCommand Class
MindFusion.Diagramming Namespace

FoldContainerCommand Methods

FoldContainerCommand.execute Method

Command.execute override.

Namespace: MindFusion.Diagramming
Assembly: FoldContainerCommand.js

Syntax

JavaScript Copy Code

```
function execute ()
```

See Also

FoldContainerCommand Members
FoldContainerCommand Class
MindFusion.Diagramming Namespace

FoldContainerCommand.redo Method

Command.redo override.

Namespace: MindFusion.Diagramming
Assembly: FoldContainerCommand.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

FoldContainerCommand Members
FoldContainerCommand Class
MindFusion.Diagramming Namespace

FoldContainerCommand.undo Method

Command.undo override.

Namespace: MindFusion.Diagramming
Assembly: FoldContainerCommand.js

Syntax

JavaScript Copy Code

```
function undo ()
```

See Also

FoldContainerCommand Members
FoldContainerCommand Class
MindFusion.Diagramming Namespace

FreeFormNode Class

A **FreeFormNode** collects all points from users' mouse or touch input and displays them as node's outline.

Syntax

JavaScript Copy Code

```
// class
FreeFormNode.prototype = {}
```

Remarks

To let users draw free-form nodes interactively, set behavior to DrawFreeForms or **LinkFreeForms**. Use the points property of **FreeFormNode** to get or set outline points programmatically. If the closed property is set, the node is drawn as a closed shape and its interior filled, or otherwise the node is drawn as a polyline. If the distance between first and last points drawn by user is shorter than autoCloseDistance, the node's **closed** property is automatically set to **true**. **autoCloseDistance** default value is Number.MAX_VALUE, so free-form nodes are always closed.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.FreeFormNode

See Also

FreeFormNode Members
MindFusion.Diagramming Namespace

FreeFormNode Members

A FreeFormNode collects all points from users' mouse or touch input and displays them as node's outline.

The following tables list the members exposed by the **FreeFormNode** type.

Public Constructors

	Name	Description
	FreeFormNode	Initializes a new instance of the **FreeFormNode** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	closed	Gets or sets whether the node's shape is closed.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)

	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)

	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	points	Gets or sets the node's outline points.
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description

	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	clone	DiagramNode.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	containsPoint	DiagramItem.containsPoint override.
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	endDrag	DiagramNode.endDrag override.
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	fromJson	DiagramNode.fromJson override.
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)

	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	getOutline	DiagramNode.getOutline override. Gets the outline of this node.
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	init	Returns a Builder object used to configure this FreeFormNode instance.
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	loadFromXml	DiagramNode.LoadFromXml override. Loads the node's content from an XML element.
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from

		DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	restoreLocationState	DiagramNode.restoreLocationState override.
	restoreState	DiagramNode.restoreState override.
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveLocationState	DiagramNode.saveLocationState override.
	saveState	DiagramNode.saveState override.
	saveToXml	Saves the node's content into an XML element.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	startDrag	DiagramNode.startDrag override.
	toJson	DiagramNode.toJson override.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	updateDrag	DiagramNode.updateDrag override.
	updateFromPoints	Updates the node's internal state after its points have changed.
S	With	Returns a Builder object used to configure and create new **FreeFormNode** instances.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)

	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object. (Inherited from DiagramNode.)
	LoadFromXml	DiagramItem.LoadFromXml override. (Inherited from DiagramNode.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)

	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from DiagramNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

FreeFormNode Constructor

Initializes a new instance of the FreeFormNode class.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function FreeFormNode ([parent])
```

Parameters

parent
Optional.
A Diagram instance whose default node attributes are copied to this node.

See Also

FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode Methods

FreeFormNode.clone Method

DiagramNode.clone override.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function clone ()
```

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.containsPoint Method

DiagramItem.containsPoint override.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function containsPoint (point)
```

Parameters

point
Point. The point to check.

Return Value

true it this node contains the specified point; otherwise, **false**.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.endDrag Method

DiagramNode.endDrag override.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function endDrag (ist)
```

Parameters

ist
An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.fromJson Method

DiagramNode.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function fromJson ()
```

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.getOutline Method

DiagramNode.getOutline override. Gets the outline of this node.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function getOutline ()
```

Return Value

Array. An array of points.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.init Method

Returns a Builder object used to configure this FreeFormNode instance.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function init ()
```

Return Value

A Builder instance.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.loadFromXml Method

DiagramNode.LoadFromXml override. Loads the node's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement

Element. An XML DOM element containing the item's data.

context

XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.restoreLocationState Method

DiagramNode.restoreLocationState override.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function restoreLocationState (ist)
```

Parameters

ist
DiagramNodeState. A DiagramNodeState instance.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.restoreState Method

DiagramNode.restoreState override.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function restoreState ()
```

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.saveLocationState Method

DiagramNode.saveLocationState override.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function saveLocationState ()
```

Return Value

DiagramNodeState. A DiagramNodeState instance.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.saveState Method

DiagramNode.saveState override.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function saveState ()
```

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.saveToXml Method

Saves the node's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement

Element. An XML DOM element that will contain the item's data.

context

XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.startDrag Method

DiagramNode.startDrag override.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function startDrag (ist)
```

Parameters

ist

An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.toJson Method

DiagramNode.toJson override.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.updateDrag Method

DiagramNode.updateDrag override.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function updateDrag (ist)
```

Parameters

ist
An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.updateFromPoints Method

Updates the node's internal state after its points have changed.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function updateFromPoints ()
```

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.With Method

Returns a Builder object used to configure and create new FreeFormNode instances.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
function With (diagram)
```

Parameters

diagram

Return Value

A Builder instance.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode Properties

FreeFormNode.closed Property

Gets or sets whether the node's shape is closed.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
get closed() {}
set closed(value) {}
```

Property Value

Boolean

true it this node has a closed shape, or **false** otherwise.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.points Property

Gets or sets the node's outline points.

Namespace: MindFusion.Diagramming
Assembly: FreeFormNode.js

Syntax

JavaScript Copy Code

```
get points() {}
set points(value) {}
```

Property Value

Array

An array of Point objects.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

GlassEffect Class

Represents a glass-like visual effect.

Syntax

JavaScript Copy Code

```
// class
GlassEffect.prototype = {}
```

Remarks

This effect can be applied to all nodes in a Diagram by adding **GlassEffect** instance to the nodeEffects collection, or to individual nodes by adding it to their effects array.

The following image illustrates the Glass effect applied to an item as well as how the different properties affect the output:



Example

The following example demonstrates how to apply glass effect to the nodes of an existing diagram.

JavaScript Copy Code

```
var GlassEffect = MindFusion.Diagramming.GlassEffect;
diagram.nodeEffects.push(new GlassEffect());
```

Inheritance Hierarchy

MindFusion.Diagramming.NodeEffect
  MindFusion.Diagramming.GlassEffect

See Also

GlassEffect Members
MindFusion.Diagramming Namespace
Node Effects
AeroEffect Class

GlassEffect Members

Represents a glass-like visual effect.

The following tables list the members exposed by the GlassEffect type.

Public Constructors

	Name	Description
	GlassEffect	Initializes a new instance of the GlassEffect class.

Public Properties

	Name	Description
	glowColor	Gets or sets the color of the glow region of the effect.
	reflectionColor	Gets or sets the color of the reflection region of the effect.
	type	Gets or sets the type of the glass effect.
	usePenAsGlow	Gets or sets whether the glow area of the effect should be painted using the stroke color of target node.

Public Methods

	Name	Description
	Clone	**ICloneable.Clone** implementation. (Inherited from NodeEffect.)

Public Events

	Name	Description
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation. (Inherited from NodeEffect.)

See Also

GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect Constructor

Initializes a new instance of the GlassEffect class.

Namespace: MindFusion.Diagramming
Assembly: NodeEffects.js

Syntax

JavaScript Copy Code

```
function GlassEffect ()
```

See Also

GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect Properties

GlassEffect.glowColor Property

Gets or sets the color of the glow region of the effect.

Namespace: MindFusion.Diagramming
Assembly: NodeEffects.js

Syntax

JavaScript Copy Code

```
get glowColor() {}
set glowColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value. The default value is "white".

See Also

GlassEffect Members
GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect.reflectionColor Property

Gets or sets the color of the reflection region of the effect.

Namespace: MindFusion.Diagramming
Assembly: NodeEffects.js

Syntax

JavaScript Copy Code

```
get reflectionColor() {}
set reflectionColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value. The default value is "white".

See Also

GlassEffect Members
GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect.type Property

Gets or sets the type of the glass effect.

Namespace: MindFusion.Diagramming
Assembly: NodeEffects.js

Syntax

JavaScript Copy Code

```
get type() {}
set type(value) {}
```

Property Value

GlassEffectType

A member of the GlassEffectType enumeration. The default is **Type1**.

See Also

GlassEffect Members
GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect.usePenAsGlow Property

Gets or sets whether the glow area of the effect should be painted using the stroke color of target node.

Namespace: MindFusion.Diagramming
Assembly: NodeEffects.js

Syntax

JavaScript Copy Code

```
get usePenAsGlow() {}
set usePenAsGlow(value) {}
```

Property Value

Boolean

true to ignore glowColor and use node's stroke color for the glow, or **false** otherwise.

See Also

GlassEffect Members
GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect Events

GlassEffectType Enumeration

Specifies the type of a GlassEffect node effect.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
GlassEffectType = {}
```

Members

	Member name	Description
	Type1	Indicates a glass effect with soft radial glow at the bottom and a sharp reflection at the top.
	Type2	Indicates a glass effect with linear glow at the bottom and sharp reflection at the top.
	Type3	Indicates a glass effect with radial glow at the bottom and smooth reflection and emphasized outline at the top.
	Type4	Indicates a glass effect with sharp reflection at the top and soft glow at the left and right sides.

Remarks

Members of this enumeration can be assigned to the type property of GlassEffect.

See Also

MindFusion.Diagramming Namespace

GridColumn Class

Represents a column in a GridPanel.

Syntax

JavaScript Copy Code

```
// class
GridColumn.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.GridColumn

See Also

GridColumn Members
MindFusion.Diagramming Namespace

GridColumn Members

Represents a column in a GridPanel.

The following tables list the members exposed by the GridColumn type.

Public Constructors

	Name	Description
	GridColumn	Initializes a new instance of the **GridColumn** class.

Public Properties

	Name	Description
	lengthType	Gets or sets how **GridPanel** determines the width of this column.
	width	Gets or sets fixed width for this column.

See Also

GridColumn Class
MindFusion.Diagramming Namespace

GridColumn Constructor

Initializes a new instance of the GridColumn class.

Namespace: MindFusion.Diagramming
Assembly: GridColumn.js

Syntax

JavaScript Copy Code

```
function GridColumn ()
```

See Also

GridColumn Class
MindFusion.Diagramming Namespace

GridColumn Properties

GridColumn.lengthType Property

Gets or sets how GridPanel determines the width of this column.

Namespace: MindFusion.Diagramming
Assembly: GridColumn.js

Syntax

JavaScript Copy Code

```
get lengthType() {}
set lengthType(value) {}
```

Property Value

LengthType

A member of the LengthType enumeration.

See Also

GridColumn Members
GridColumn Class
MindFusion.Diagramming Namespace

GridColumn.width Property

Gets or sets fixed width for this column.

Namespace: MindFusion.Diagramming
Assembly: GridColumn.js

Syntax

JavaScript Copy Code

```
get width() {}
set width(value) {}
```

Property Value

Number

A number specifying the column's width.

See Also

GridColumn Members
GridColumn Class
MindFusion.Diagramming Namespace

GridPanel Class

A layout container that arranges its child components in a grid.

Syntax

JavaScript Copy Code

```
// class
GridPanel.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.Container
  MindFusion.Diagramming.Panel
    MindFusion.Diagramming.StackPanel

See Also

GridPanel Members
MindFusion.Diagramming Namespace

GridPanel Members

A layout container that arranges its child components in a grid.

The following tables list the members exposed by the GridPanel type.

Public Constructors

	Name	Description
	GridPanel	Initializes a new instance of the GridPanel class.

Public Properties

	Name	Description
	columns	Gets a list of GridColumn objects specifying column attributes.
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	rows	Gets a list of GridRow objects specifying row attributes.
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

Public Methods

	Name	Description
	getComponents	Gets the child components of this panel. (Inherited from Panel.)

See Also

GridPanel Constructor

Initializes a new instance of the GridPanel class.

Namespace: MindFusion.Diagramming
Assembly: GridPanel.js

Syntax

JavaScript Copy Code

```
function GridPanel ()
```

See Also

GridPanel Class
MindFusion.Diagramming Namespace

GridPanel Properties

GridPanel.columns Property

Gets a list of GridColumn objects specifying column attributes.

Namespace: MindFusion.Diagramming
Assembly: GridPanel.js

Syntax

JavaScript Copy Code

```
get columns() {}
```

Property Value

Array

An array containing the grid columns.

See Also

GridPanel Members
GridPanel Class
MindFusion.Diagramming Namespace

GridPanel.rows Property

Gets a list of GridRow objects specifying row attributes.

Namespace: MindFusion.Diagramming
Assembly: GridPanel.js

Syntax

JavaScript Copy Code

```
get rows() {}
```

Property Value

Array

An array containing the grid rows.

See Also

GridPanel Members
GridPanel Class
MindFusion.Diagramming Namespace

GridRouter Class

Implements a link routing algorithm that uses a grid to represent the free and occupied regions in the diagram.

Syntax

JavaScript Copy Code

```
// class
GridRouter.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Router
MindFusion.Diagramming.GridRouter

See Also

GridRouter Members
MindFusion.Diagramming Namespace

GridRouter Members

Implements a link routing algorithm that uses a grid to represent the free and occupied regions in the diagram.

The following tables list the members exposed by the GridRouter type.

Public Constructors

	Name	Description
	GridRouter	Initializes a new instance of the GridRouter class.

Public Properties

	Name	Description
	avoidOverlaps	Gets or sets a value indicating whether to avoid overlapping links. (Inherited from Router.)
	crossingCost	Gets or sets a value added to the total cost of a route which causes a link to cross another link.
	gridSize	Gets or sets the size of the pieces of the routing grid.
	lengthCost	Gets or sets a value added to the total cost of a route for each piece of the routing grid occupied by the route.
	nodeVicinityCost	Gets or sets a value added to the total cost of a route if the route passes too close to a node.
	nodeVicinitySize	Gets or sets the size of buffer zone around nodes for which routes get a penalty cost if they pass through it.
	turnCost	Gets or sets a value added to the total cost of a route if the route passes too close to a node.

Public Methods

	Name	Description
	route	Called when a single link must be routed. (Inherited from Router.)
	routeInternal	Calculates actual route and returns it as an array of points. (Inherited from Router.)
	routeLinks	Called when more than one link must be routed. (Inherited from Router.)

See Also

GridRouter Class
MindFusion.Diagramming Namespace

GridRouter Constructor

Initializes a new instance of the GridRouter class.

Namespace: MindFusion.Diagramming
Assembly: GridRouter.js

Syntax

JavaScript Copy Code

```
function GridRouter ([diagram])
```

Parameters

diagram
Optional.
Diagram. Gets or sets the diagram this router is associated with.

See Also

GridRouter Class
MindFusion.Diagramming Namespace

GridRouter Properties

GridRouter.crossingCost Property

Gets or sets a value added to the total cost of a route which causes a link to cross another link.

Namespace: MindFusion.Diagramming
Assembly: GridRouter.js

Syntax

JavaScript Copy Code

```
get crossingCost() {}
set crossingCost(value) {}
```

Property Value

Number

A number specifying the crossing cost. The default is 4.

Remarks

This is a penalty value added to the total cost of an evaluated link route if it crosses another link. The routing algorithm aims to find paths that have a minimal cost, so assigning a greater value to **crossingCost** leads to fewer link intersections. This value is

considered only when routing multiple links at once, e.g. by routeAllLinks method.

See Also

GridRouter Members
GridRouter Class
MindFusion.Diagramming Namespace

GridRouter.gridSize Property

Gets or sets the size of the pieces of the routing grid.

Namespace: MindFusion.Diagramming
Assembly: GridRouter.js

Syntax

JavaScript Copy Code

```
get gridSize() {}
set gridSize(value) {}
```

Property Value

Number

A number specifying the size of the pieces of the routing grid. The default is 4.

See Also

GridRouter Members
GridRouter Class
MindFusion.Diagramming Namespace

GridRouter.lengthCost Property

Gets or sets a value added to the total cost of a route for each piece of the routing grid occupied by the route.

Namespace: MindFusion.Diagramming
Assembly: GridRouter.js

Syntax

JavaScript Copy Code

```
get lengthCost() {}
set lengthCost(value) {}
```

Property Value

Number

A number specifying the length cost. The default is 2.

See Also

GridRouter Members
GridRouter Class
MindFusion.Diagramming Namespace

GridRouter.nodeVicinityCost Property

Gets or sets a value added to the total cost of a route if the route passes too close to a node.

Namespace: MindFusion.Diagramming
Assembly: GridRouter.js

Syntax

JavaScript Copy Code

```
get nodeVicinityCost() {}
set nodeVicinityCost(value) {}
```

Property Value

Number

A number specifying the node vicinity cost. The default is 2.

See Also

GridRouter Members
GridRouter Class
MindFusion.Diagramming Namespace

GridRouter.nodeVicinitySize Property

Gets or sets the size of buffer zone around nodes for which routes get a penalty cost if they pass through it.

Namespace: MindFusion.Diagramming
Assembly: GridRouter.js

Syntax

JavaScript Copy Code

```
get nodeVicinitySize() {}
set nodeVicinitySize(value) {}
```

Property Value

Number

A number specifying the size of the buffer area in which passing routes receive a penalty cost. The default is 12.

See Also

GridRouter Members
GridRouter Class
MindFusion.Diagramming Namespace

GridRouter.turnCost Property

Gets or sets a value added to the total cost of a route if the route passes too close to a node.

Namespace: MindFusion.Diagramming
Assembly: GridRouter.js

Syntax

JavaScript Copy Code

```
get turnCost() {}
set turnCost(value) {}
```

Property Value

Number

A number specifying the node vicinity cost. The default is 2.

See Also

GridRouter Members
GridRouter Class
MindFusion.Diagramming Namespace

GridRow Class

Represents a row in a GridPanel.

Syntax

JavaScript Copy Code

```
// class
GridRow.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.GridRow

See Also

GridRow Members
MindFusion.Diagramming Namespace

GridRow Members

Represents a row in a GridPanel.

The following tables list the members exposed by the GridRow type.

Public Constructors

	Name	Description
	GridRow	Initializes a new instance of the **GridRow** class.

Public Properties

	Name	Description
	height	Gets or sets fixed height for this row.
	lengthType	Gets or sets how GridPanel determines the height of this row.

See Also

GridRow Class
MindFusion.Diagramming Namespace

GridRow Constructor

Initializes a new instance of the GridRow class.

Namespace: MindFusion.Diagramming
Assembly: GridRow.js

Syntax

JavaScript Copy Code

```
function GridRow ()
```

See Also

GridRow Class
MindFusion.Diagramming Namespace

GridRow Properties

GridRow.height Property

Gets or sets fixed height for this row.

Namespace: MindFusion.Diagramming
Assembly: GridRow.js

Syntax

JavaScript Copy Code

```
get height() {}
set height(value) {}
```

Property Value

Number

A number specifying the rows's height.

See Also

GridRow Members
GridRow Class
MindFusion.Diagramming Namespace

GridRow.lengthType Property

Gets or sets how GridPanel determines the height of this row.

Namespace: MindFusion.Diagramming
Assembly: GridRow.js

Syntax

JavaScript Copy Code

```
get lengthType() {}
set lengthType(value) {}
```

Property Value

LengthType

A member of the LengthType enumeration.

See Also

GridRow Members
GridRow Class
MindFusion.Diagramming Namespace

GridStyle Enumeration

Defines values that specify the visual style of the alignment grid.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
GridStyle = {}
```

Members

	Member name	Description
	Points	The grid is painted as a matrix of points.
	Lines	The grid is represented as a series of crossing horizontal and vertical lines.
	Crosses	The grid is painted as a matrix of crosses.

Remarks

Members of this enumeration can be assigned to the gridStyle property of Diagram.

See Also

MindFusion.Diagramming Namespace

HandlesStyle Enumeration

Enumerates possible visual styles for frames and handles drawn around selected nodes.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
HandlesStyle = {}
```

Members

	Member name	Description
	Invisible	There is no visible difference between selected and unselected node.
	SquareHandles	If selected, the node is drawn with square modification handles.
	DashFrame	If selected, a node is drawn with a dashed frame. The frame enables resizing the node. The center of the node allows moving it and the edge area enables creating links.
	HatchFrame	If selected, the item is drawn with hatched frame. Modifications can be done as with DashFrame.
	HatchHandles	If selected, a node is drawn with both hatched frame and modification handles. The frame enable moving the node, and the handles enable resizing. Links are created from any point of the interior.
	HatchHandles2	Looks like HatchHandles, but allows moving the node from the interior area. Links are created from the points near the edge.
	HatchHandles3	Behaves like HatchHandles2, but paints the selection frame using a denser brush pattern.
	MoveOnly	If a node is selected, it can be only moved. Best used with ModificationStart set to AutoHandles.

	EasyMove	Any point of the interior of a node enables moving the node, except a small area in the center that allows creating links. Best used with ModificationStart set to AutoHandles.
	SquareHandles2	Square resize handles are drawn around a selected node. There isn't a move handle in the center as with the SquareHandles style, but nodes can be moved by dragging their interior or caption areas. Links can be created by drawing from near the edges of a node.
	Custom	The control raises the drawAdjustmentHandles and hitTestAdjustmentHandles events to let your application perform custom drawing and hit testing of selection handles.
	InvisibleMove	There aren't any selection handles displayed. The selected node can be only moved, but not resized.
	RoundAndSquare	Corner resize handles are round, remaining handles are square.
	RoundAndSquare2	Corner resize handles are round, remaining handles are square. In addition, there is a dashed frame drawn around selected nodes.

Remarks

Members of this enumeration can be assigned to the handlesStyle property of DiagramNode.

See Also

MindFusion.Diagramming Namespace

HandlesVisualStyle Class

Defines the pen and brush objects used to draw items' adjustment handles.

Syntax

JavaScript Copy Code

```
// class
HandlesVisualStyle.prototype = {}
```

Remarks

The component uses several different **HandlesVisualStyle** objects depending on the context. They can be accessed through the activeItemHandlesStyle, selectedItemHandlesStyle and disabledHandlesStyle properties of Diagram.

Inheritance Hierarchy

MindFusion.Diagramming.HandlesVisualStyle

See Also

HandlesVisualStyle Members
MindFusion.Diagramming Namespace

HandlesVisualStyle Members

Defines the pen and brush objects used to draw items' adjustment handles.

The following tables list the members exposed by the HandlesVisualStyle type.

Public Constructors

	Name	Description
	HandlesVisualStyle	Initializes a new instance of the HandlesVisualStyle class.

Public Properties

	Name	Description
	controlPointBrush	Gets or sets the brush used to fill the interior of shape control points.
	dashStroke	Gets or sets the color used to stroke dashed selection frames.
	dashStrokeThickness	Gets or sets the line width applied when stroking dashed selection frames.
	handleBrush	Gets or sets the brush used to fill the interior of selection handles.
	handleStroke	Gets or sets the color used to stroke borders of adjustment handles.
	handleStrokeThickness	Gets or sets the line width applied when stroking borders of adjustment handles.
	hatchBrush	Gets or sets the brush used to draw hatch frames.
	patternBrush	Gets or sets the brush used to draw selection frames for the MoveOnly style.
	rotationHandleBrush	Gets or sets the brush used to fill the interior of the rotation handle.

See Also

HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle Constructor

Initializes a new instance of the HandlesVisualStyle class.

Namespace: MindFusion.Diagramming
Assembly: HandlesVisualStyle.js

Syntax

JavaScript Copy Code

```
function HandlesVisualStyle ()
```

See Also

HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle Properties

HandlesVisualStyle.controlPointBrush Property

Gets or sets the brush used to fill the interior of shape control points.

Namespace: MindFusion.Diagramming
Assembly: HandlesVisualStyle.js

Syntax

JavaScript Copy Code

```
get controlPointBrush() {}
set controlPointBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.dashStroke Property

Gets or sets the color used to stroke dashed selection frames.

Namespace: MindFusion.Diagramming
Assembly: HandlesVisualStyle.js

Syntax

JavaScript Copy Code

```
get dashStroke() {}
set dashStroke(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.dashStrokeThickness Property

Gets or sets the line width applied when stroking dashed selection frames.

Namespace: MindFusion.Diagramming
Assembly: HandlesVisualStyle.js

Syntax

JavaScript Copy Code

```
get dashStrokeThickness() {}
set dashStrokeThickness(value) {}
```

Property Value

Number

A number specifying stroke thickness.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.handleBrush Property

Gets or sets the brush used to fill the interior of selection handles.

Namespace: MindFusion.Diagramming
Assembly: HandlesVisualStyle.js

Syntax

JavaScript Copy Code

```
get handleBrush() {}
set handleBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.handleStroke Property

Gets or sets the color used to stroke borders of adjustment handles.

Namespace: MindFusion.Diagramming
Assembly: HandlesVisualStyle.js

Syntax

JavaScript Copy Code

```
get handleStroke() {}
set handleStroke(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.handleStrokeThickness Property

Gets or sets the line width applied when stroking borders of adjustment handles.

Namespace: MindFusion.Diagramming
Assembly: HandlesVisualStyle.js

Syntax

JavaScript Copy Code

```
get handleStrokeThickness() {}
set handleStrokeThickness(value) {}
```

Property Value

Number

A number specifying stroke thickness.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.hatchBrush Property

Gets or sets the brush used to draw hatch frames.

Namespace: MindFusion.Diagramming
Assembly: HandlesVisualStyle.js

Syntax

JavaScript Copy Code

```
get hatchBrush() {}
set hatchBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.patternBrush Property

Gets or sets the brush used to draw selection frames for the MoveOnly style.

Namespace: MindFusion.Diagramming
Assembly: HandlesVisualStyle.js

Syntax

JavaScript Copy Code

```
get patternBrush() {}
set patternBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.rotationHandleBrush Property

Gets or sets the brush used to fill the interior of the rotation handle.

Namespace: MindFusion.Diagramming
Assembly: HandlesVisualStyle.js

Syntax

JavaScript Copy Code

```
get rotationHandleBrush() {}
set rotationHandleBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HeaderEventArgs Class

Contains the arguments passed to handlers of header-related events.

Syntax

JavaScript Copy Code

```
// class
HeaderEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.EventArgs
  MindFusion.Diagramming.HeaderEventArgs
    MindFusion.Diagramming.HeaderResizeEventArgs

See Also

HeaderEventArgs Members
MindFusion.Diagramming Namespace
Events Class

HeaderEventArgs Members

Contains the arguments passed to handlers of header-related events.

The following tables list the members exposed by the HeaderEventArgs type.

Public Constructors

	Name	Description
	HeaderEventArgs	Initializes a new instance of the HeaderEventArgs class.

Public Properties

	Name	Description
	handled	Gets or sets a value indicating whether the event has been handled. (Inherited from EventArgs.)
	header	Gets the header related to the event.

See Also

HeaderEventArgs Class
MindFusion.Diagramming Namespace

HeaderEventArgs Constructor

Initializes a new instance of the HeaderEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function HeaderEventArgs ()
```

See Also

HeaderEventArgs Class
MindFusion.Diagramming Namespace

HeaderEventArgs Properties

HeaderEventArgs.header Property

Gets the header related to the event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get header() {}
```

Property Value

Header

The grid header.

See Also

HeaderEventArgs Members
HeaderEventArgs Class
MindFusion.Diagramming Namespace

HeaderResizeEventArgs Class

Contains the arguments passed to handlers of header resize events.

Syntax

JavaScript Copy Code

```
// class
HeaderResizeEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.EventArgs
  MindFusion.Diagramming.HeaderEventArgs
    MindFusion.Diagramming.HeaderResizeEventArgs

See Also

HeaderResizeEventArgs Members

Contains the arguments passed to handlers of header resize events.

The following tables list the members exposed by the HeaderResizeEventArgs type.

Public Constructors

	Name	Description
	HeaderResizeEventArgs	Initializes a new instance of the HeaderEventArgs class.

Public Properties

	Name	Description
	cancel	Gets or sets a value indicating whether to allow the current operation.
	handled	Gets or sets a value indicating whether the event has been handled. (Inherited from EventArgs.)
	header	Gets the header related to the event. (Inherited from HeaderEventArgs.)
	horizontal	Gets a value indicating whether the user resizes the header horizontally.

Public Methods

	Name	Description
	cancelDrag	Immediately cancels the resize operation.

See Also

HeaderResizeEventArgs Class
MindFusion.Diagramming Namespace

HeaderResizeEventArgs Constructor

Initializes a new instance of the HeaderEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function HeaderResizeEventArgs ()
```

See Also

HeaderResizeEventArgs Class
MindFusion.Diagramming Namespace

HeaderResizeEventArgs Methods

HeaderResizeEventArgs.cancelDrag Method

Immediately cancels the resize operation.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function cancelDrag ()
```

See Also

HeaderResizeEventArgs Members
HeaderResizeEventArgs Class
MindFusion.Diagramming Namespace

HeaderResizeEventArgs Properties

HeaderResizeEventArgs.cancel Property

Gets or sets a value indicating whether to allow the current operation.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get cancel() {}
set cancel(value) {}
```

Property Value

Boolean

true to cancel the operation, or **false** otherwise.

See Also

HeaderResizeEventArgs Members
HeaderResizeEventArgs Class
MindFusion.Diagramming Namespace

HeaderResizeEventArgs.horizontal Property

Gets a value indicating whether the user resizes the header horizontally.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get horizontal() {}
```

Property Value

Boolean

true if resizing horizontally, or **false** otherwise.

See Also

HeaderResizeEventArgs Members
HeaderResizeEventArgs Class
MindFusion.Diagramming Namespace

InplaceEditEventArgs Class

Contains the arguments passed to handlers of inplace edit related events.

Syntax

JavaScript Copy Code

```
// class
InplaceEditEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.EventArgs
  MindFusion.Diagramming.InplaceEditEventArgs

See Also

InplaceEditEventArgs Members
MindFusion.Diagramming Namespace
Events Class

InplaceEditEventArgs Members

Contains the arguments passed to handlers of inplace edit related events.

The following tables list the members exposed by the InplaceEditEventArgs type.

Public Constructors

	Name	Description
	InplaceEditEventArgs	Initializes a new instance of the **InplaceEditEventArgs** class.

Public Properties

	Name	Description
	bounds	Gets the bounds of the item being edited.
	control	Gets or sets the control used to edit the item's text.
	handled	Gets or sets a value indicating whether the event has been handled. (Inherited from EventArgs.)
	item	Gets a reference to the item being edited.

See Also

InplaceEditEventArgs Class
MindFusion.Diagramming Namespace

InplaceEditEventArgs Constructor

Initializes a new instance of the InplaceEditEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function InplaceEditEventArgs ()
```

See Also

InplaceEditEventArgs Class
MindFusion.Diagramming Namespace

InplaceEditEventArgs Properties

InplaceEditEventArgs.bounds Property

Gets the bounds of the item being edited.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get bounds() {}
```

Property Value

Rect

A Rect instance containing item's coordinates.

See Also

InplaceEditEventArgs Members
InplaceEditEventArgs Class
MindFusion.Diagramming Namespace

InplaceEditEventArgs.control Property

Gets or sets the control used to edit the item's text.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get control() {}
set control(value) {}
```

Property Value

Element

An HTML DOM element to use as editor.

See Also

InplaceEditEventArgs Members
InplaceEditEventArgs Class
MindFusion.Diagramming Namespace

InplaceEditEventArgs.item Property

Gets a reference to the item being edited.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript — Copy Code

```
get item() {}
```

Property Value

DiagramItem

A DiagramItem instance.

See Also

InplaceEditEventArgs Members
InplaceEditEventArgs Class
MindFusion.Diagramming Namespace

Interaction Class

Encapsulates the current state of user interaction with the control.

Syntax

JavaScript — Copy Code

```
// class
Interaction.prototype = {}
```

Inheritance Hierarchy

See Also

Interaction Members
MindFusion.Diagramming Namespace

Interaction Members

Encapsulates the current state of user interaction with the control.

The following tables list the members exposed by the Interaction type.

Public Constructors

	Name	Description
	Interaction	Initializes a new instance of the **Interaction** class.

See Also

Interaction Class
MindFusion.Diagramming Namespace

Interaction Constructor

Initializes a new instance of the Interaction class.

Namespace: MindFusion.Diagramming
Assembly: Interaction.js

Syntax

```
JavaScript                                                    Copy Code
function Interaction ([diagram])
```

Parameters

diagram
Optional.

See Also

Interaction Class
MindFusion.Diagramming Namespace

InteractionState Class

Encapsulates the current state of user interaction with the control.

Syntax

```
JavaScript                                                    Copy Code
// class
InteractionState.prototype = {}
```

Inheritance Hierarchy

See Also

InteractionState Members
MindFusion.Diagramming Namespace

InteractionState Members

Encapsulates the current state of user interaction with the control.

The following tables list the members exposed by the InteractionState type.

Public Constructors

	Name	Description
	InteractionState	Initializes a new instance of the **InteractionState** class.

Public Methods

	Name	Description
	cancel	Cancels the user interaction. Call this method to cancel interaction, usually in response to the user clicking the right mouse button while dragging.
	commit	Invoked to commit the changes made by the user.
	isAllowed	Checks whether the current interaction can be completed at the specified point.
	start	This method is invoked in the beginning of the interaction, usually when the user presses the left mouse button over a modification handle.

	update	This method is invoked to update the interaction state, usually in response to the user moving the mouse after the modification has started.

See Also

InteractionState Class
MindFusion.Diagramming Namespace

InteractionState Constructor

Initializes a new instance of the InteractionState class.

Namespace: MindFusion.Diagramming
Assembly: InteractionState.js

Syntax

JavaScript Copy Code

```
function InteractionState (modifiedItem, action, adjustmentHandle, point)
```

Parameters

modifiedItem
The DiagramItem to be created or modified.

action
One of the Action enumeration values, specifying what kind of action to perform.

adjustmentHandle
The index of the selection handle to be dragged with the mouse, defines what kind of modification to perform.

point
A Point specifying the mouse position where the user starts to draw or modify an item.

See Also

InteractionState Class
MindFusion.Diagramming Namespace

InteractionState Methods

InteractionState.cancel Method

Cancels the user interaction. Call this method to cancel interaction, usually in response to the user clicking the right mouse button while dragging.

Namespace: MindFusion.Diagramming
Assembly: InteractionState.js

Syntax

JavaScript Copy Code

```
function cancel (diagram)
```

Parameters

diagram
The Diagram, at which the action is performed.

Remarks

Call this method to cancel interaction, usually in response to the user clicking the right mouse button while dragging.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.commit Method

Invoked to commit the changes made by the user.

Namespace: MindFusion.Diagramming
Assembly: InteractionState.js

Syntax

JavaScript Copy Code

```
function commit (point)
```

Parameters

point
A Point instance indicating the location of the cursor, in document coordinates.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.isAllowed Method

Checks whether the current interaction can be completed at the specified point.

Namespace: MindFusion.Diagramming
Assembly: InteractionState.js

Syntax

JavaScript Copy Code

```
function isAllowed (point)
```

Parameters

point
A Point instance indicating the location of the cursor, in document coordinates.

Return Value

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.start Method

This method is invoked in the beginning of the interaction, usually when the user presses the left mouse button over a modification handle.

Namespace: MindFusion.Diagramming
Assembly: InteractionState.js

Syntax

JavaScript Copy Code

```
function start ()
```

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.update Method

This method is invoked to update the interaction state, usually in response to the user moving the mouse after the modification has started.

Namespace: MindFusion.Diagramming
Assembly: InteractionState.js

Syntax

JavaScript Copy Code

```
function update (point)
```

Parameters

point
A Point instance indicating the location of the cursor, in document coordinates.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

ItemEventArgs Class

Contains the arguments passed to handlers of item-related events.

Syntax

JavaScript Copy Code

```
// class
ItemEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.CancelEventArgs
  MindFusion.Diagramming.ItemEventArgs

See Also

ItemEventArgs Members
MindFusion.Diagramming Namespace

ItemEventArgs Members

Contains the arguments passed to handlers of item-related events.

The following tables list the members exposed by the ItemEventArgs type.

Public Constructors

	Name	Description

	ItemEventArgs	Initializes a new instance of the ItemEventArgs class.

Public Properties

	Name	Description
	adjustmentHandle	Gets the adjustment handle being moved by the user.
	cancel	Gets or sets a value indicating whether to allow the current operation. (Inherited from CancelEventArgs.)
	context	Gets the canvas rendering context.
	handled	Gets or sets a value indicating whether the event has been handled. (Inherited from EventArgs.)
	item	Gets the diagram item for which the event is raised.
	mouseButton	Gets which mouse button has been pressed.
	mousePosition	Gets the current mouse pointer position.
	newText	Gets the text that has just been entered by the user.
	oldText	Gets the text that had been displayed in the node before the user edited it.

Public Methods

	Name	Description
	cancelDrag	Immediately cancels the creation or modification of the node.

See Also

ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemEventArgs Constructor

Initializes a new instance of the ItemEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function ItemEventArgs (args)
```

Parameters

args

See Also

ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemEventArgs Methods

ItemEventArgs.cancelDrag Method

Immediately cancels the creation or modification of the node.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function cancelDrag ()
```

See Also

ItemEventArgs Members
ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemEventArgs Properties

ItemEventArgs.adjustmentHandle Property

Gets the adjustment handle being moved by the user.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get adjustmentHandle() {}
set adjustmentHandle(value) {}
```

Property Value

AdjustmentHandles

A member of the AdjustmentHandles enumeration.

See Also

ItemEventArgs Members
ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemEventArgs.context Property

Gets the canvas rendering context.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get context() {}
```

Property Value

CanvasRenderingContext2D

A rendering context used to draw on the Canvas element.

See Also

ItemEventArgs Members
ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemEventArgs.item Property

Gets the diagram item for which the event is raised.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript — Copy Code

```
get item() {}
```

Property Value

DiagramItem

An instance of the DiagramItem class.

See Also

ItemEventArgs Members
ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemEventArgs.mouseButton Property

Gets which mouse button has been pressed.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript — Copy Code

```
get mouseButton() {}
```

Property Value

MouseButton

A member of the MouseButton enumeration.

See Also

ItemEventArgs Members
ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemEventArgs.mousePosition Property

Gets the current mouse pointer position.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript — Copy Code

```
get mousePosition() {}
```

Property Value

Point

A Point instance containing mouse pointer coordinates.

See Also

ItemEventArgs Members
ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemEventArgs.newText Property

Gets the text that has just been entered by the user.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get newText() {}
```

Property Value

String

A string containing newly entered text.

See Also

ItemEventArgs Members
ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemEventArgs.oldText Property

Gets the text that had been displayed in the node before the user edited it.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get oldText() {}
```

Property Value

String

A string containing item's original text.

See Also

ItemEventArgs Members
ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemLabel Class

Represents a label displayed by a DiagramItem.

Syntax

JavaScript Copy Code

```
// class
ItemLabel.prototype = {}
```

Remarks

This class represents supplementary labels displayed by an item, in addition to its main text property. Call the addLabel method to create a label, and customize it through various members of type-specific LinkLabel and NodeLabel classes.

A label can also display an image assigned to the image or imageLocation property. This lets you draw multiple icons per node, or along a link's geometry.

Inheritance Hierarchy

MindFusion.Diagramming.ItemLabel

See Also

ItemLabel Members
MindFusion.Diagramming Namespace

ItemLabel Members

Represents a label displayed by a DiagramItem.

The following tables list the members exposed by the ItemLabel type.

Public Constructors

	Name	Description
	ItemLabel	Initializes a new instance of the **ItemLabel** class.

Public Properties

	Name	Description
	brush	Gets or sets the brush used to draw the label's background.
	contentPadding	Gets or sets the distance between image and text when both are specified.
	enableStyledText	Gets or sets a value indicating whether styled text rendering is enabled.
	enableWrap	Gets or sets a value indicating whether text wrapping is enabled.
	font	Gets or sets the font used to render label's text.
	horizontalAlign	Gets or sets the label's horizontal alignment.
	horizontalOffset	Gets or sets label's horizontal offset from its pin point.
	image	Gets or sets the image displayed by this label.
	imageAlign	Gets or sets the alignment of label's image relative to text.
	imageContent	Gets or sets a Base64-encoded image to display inside this label.
	imageLocation	Gets or sets URL of the image to display inside this label.
	imageSize	Gets or sets a size that overrides the native image size.
	locked	Gets or sets a value indicating whether users are allowed to edit this label.
	margin	Gets or sets the label's margin.
	maxWidth	Gets or sets the maximum width of this label.
	text	Gets or sets the label's text.

	textColor	Gets or sets the text color.
	verticalAlign	Gets or sets the label's vertical alignment.
	verticalOffset	Gets or sets label's vertical offset from its pin point.

See Also

ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel Constructor

Initializes a new instance of the ItemLabel class.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
function ItemLabel (item, text)
```

Parameters

item
DiagramItem. The DiagramItem that displays the label.

text
String. The label's text.

See Also

ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel Properties

ItemLabel.brush Property

Gets or sets the brush used to draw the label's background.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get brush() {}
set brush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.contentPadding Property

Gets or sets the distance between image and text when both are specified.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get contentPadding() {}
set contentPadding(value) {}
```

Property Value

Number

A Number specifying the distance.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.enableStyledText Property

Gets or sets a value indicating whether styled text rendering is enabled.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get enableStyledText() {}
set enableStyledText(value) {}
```

Property Value

Boolean

true to parse the label's text in order to find formatting tags, or **false** otherwise.

Remarks

Styled text recognizes the following format tags:

Tag	Description
text	Indicates italic text.
text	Indicates bold text.
<u>text</u>	Indicates underlined text.
_{text}	Indicates subscript.
^{text}	Indicates superscript.
<color=#RRGGBB>text</color>	Changes the color of the text. The red, green and blue color channels are specified as two-digit hexadecimal numbers.
 	Indicates line break.
<a=url>text	Renders specified text as a hyperlink and raises hyperlinkClicked event when link has been clicked.
<size=N>text</size>	Changes the font size.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.enableWrap Property

Gets or sets a value indicating whether text wrapping is enabled.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get enableWrap() {}
set enableWrap(value) {}
```

Property Value

Boolean

true to wrap the label's text, or **false** otherwise.

Remarks

In order to wrap text, you must also set the maxWidth property to specify maximum line length.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.font Property

Gets or sets the font used to render label's text.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get font() {}
set font(value) {}
```

Property Value

Font

An instance of the Font class.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.horizontalAlign Property

Gets or sets the label's horizontal alignment.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get horizontalAlign() {}
set horizontalAlign(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.horizontalOffset Property

Gets or sets label's horizontal offset from its pin point.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get horizontalOffset() {}
set horizontalOffset(value) {}
```

Property Value

Number

A number specifying label's horizontal position, specified in diagram's measureUnit.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.image Property

Gets or sets the image displayed by this label.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get image() {}
set image(value) {}
```

Property Value

HTMLImageElement

An image object to render inside the label.

Remarks

If a label displays both image and text, the image position relative to text depends on the imageAlign property.

If the label is only intended to display an icon image and no text, you could also set its locked property to prevent users from entering text.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace
imageSize Property
imageLocation Property

ItemLabel.imageAlign Property

Gets or sets the alignment of label's image relative to text.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get imageAlign() {}
set imageAlign(value) {}
```

Property Value

ImageAlign

A member of the ImageAlign enumeration. The default value is **MiddleLeft**.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace
image Property
imageLocation Property

ItemLabel.imageContent Property

Gets or sets a Base64-encoded image to display inside this label.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get imageContent() {}
set imageContent(value) {}
```

Property Value

String

A Base64-encoded representation of an image.

Remarks

If a label displays both image and text, the image position relative to text depends on the imageAlign property.

If the label is only intended to display an icon image and no text, you could also set its locked property to prevent users from entering text.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.imageLocation Property

Gets or sets URL of the image to display inside this label.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript — Copy Code

```
get imageLocation() {}
set imageLocation(value) {}
```

Property Value

String

A URL string that identifies the image to be loaded.

Remarks

If a label displays both image and text, the image position relative to text depends on the imageAlign property.

If the label is only intended to display an icon image and no text, you could also set its locked property to prevent users from entering text.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace
image Property

ItemLabel.imageSize Property

Gets or sets a size that overrides the native image size.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript — Copy Code

```
get imageSize() {}
set imageSize(value) {}
```

Property Value

Size

A Size instance.

Remarks

By default, image is rendered using bitmap's intrinsic size. Set this value to increase or reduce the size of the rendered image.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace
image Property
imageLocation Property

ItemLabel.locked Property

Gets or sets a value indicating whether users are allowed to edit this label.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get locked() {}
set locked(value) {}
```

Property Value

Boolean

true to prevent users from editing this label, or **false** otherwise.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.margin Property

Gets or sets the label's margin.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get margin() {}
set margin(value) {}
```

Property Value

Thickness

A Thickness instance specifying the margin space on all sides of label.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.maxWidth Property

Gets or sets the maximum width of this label.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get maxWidth() {}
set maxWidth(value) {}
```

Property Value

Number

The maximum width.

Remarks

This value is used when enableWrap is set to **true**.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.text Property

Gets or sets the label's text.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get text() {}
set text(value) {}
```

Property Value

String

A string containing the label's text.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.textColor Property

Gets or sets the text color.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

A string value identifying the text color.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.verticalAlign Property

Gets or sets the label's vertical alignment.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get verticalAlign() {}
set verticalAlign(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.verticalOffset Property

Gets or sets label's vertical offset from its pin point.

Namespace: MindFusion.Diagramming
Assembly: ItemLabel.js

Syntax

JavaScript Copy Code

```
get verticalOffset() {}
set verticalOffset(value) {}
```

Property Value

Number

A number specifying label's vertical position, specified in diagram's measureUnit.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemListView Class

A list view control that hosts DiagramItems and supports drag-and-drop to DiagramView.

Syntax

JavaScript Copy Code

```
// class
ItemListView.prototype = {}
```

Remarks

Call the addItem method to add an item to the view.

Drag-and-drop operations create a copy of the dragged item, and set its size to newInstanceSize. The copy is created by calling item's clone method. You can override **clone** to copy property values of custom item subclasses.

Icons in the list are drawn by scaling **newInstanceSize** down to iconSize. E.g. if a list icon is half the size of the diagram item it creates, its graphic elements such as cells, labels, strokes would be rendered at half scale too (as opposed to rendering with same label / stroke sizes but in smaller bounding rectangle). The scaling code assumes measureUnit of the target diagram and the list view have matching values.

Inheritance Hierarchy

MindFusion.Controls.CanvasControl
  MindFusion.Diagramming.ListViewBase

MindFusion.Diagramming.ItemListView

See Also

ItemListView Members
MindFusion.Diagramming Namespace
Palette Class

ItemListView Members

A list view control that hosts DiagramItems and supports drag-and-drop to DiagramView.

The following tables list the members exposed by the ItemListView type.

Public Constructors

	Name	Description
	ItemListView	Creates an instance of **ItemListView**.

Public Properties

	Name	Description
	autoConnectDistance	Gets or sets the maximum distance to a node when auto connecting links.
	autoConnectLinks	Gets or sets whether links should be automatically connected to the nearest node.
	backColor	Gets or sets the background color of the list view. (Inherited from ListViewBase.)
	columns	Gets or sets the number of columns in control's layout.
	draggedItem	Gets the currently dragged item. (Inherited from ListViewBase.)
	element	Returns a reference to the control's DOM element. (Inherited from Control.)
	enabled	Gets or sets whether mouse events are enabled. (Inherited from Control.)
	iconSize	Gets or sets the size of items displayed in this list view. (Inherited from ListViewBase.)
	items	Gets the items contained in this list view. (Inherited from ListViewBase.)
	labelPosition	Gets or sets a value indicating the position of labels relative to item icons.
	licenseLocation	Gets or sets the location of the license key file. (Inherited from Control.)
	measureUnit	Gets the unit of measure used for logical coordinates. (Inherited from CanvasControl.)
	newInstanceSize	Gets or sets the default size of items created by drag-and-drop from this list view. (Inherited from ListViewBase.)
	orientation	Gets or sets the orientation of the view layout. (Inherited from ListViewBase.)
	padding	Gets or sets the padding space between item's slot bounds and its contents. (Inherited from ListViewBase.)
	shapeLibraryLocation	Gets the location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path. (Inherited from ListViewBase.)
	textColor	Gets or sets the color used to paint item captions. (Inherited from ListViewBase.)

	tooltipDelay	Gets or sets the delay before a tooltip is shown. (Inherited from ListViewBase.)

Public Methods

	Name	Description
	addEventListener	Registers a single event listener on the instance. (Inherited from Disposable.)
S	addHandlers	Adds an event listener to a DOM element. (Inherited from Control.)
	addItem	Adds an item to the list view. (Inherited from ListViewBase.)
	clearAll	Removes all items from the list view. (Inherited from ListViewBase.)
S	clearHandlers	Removes all event listeners from a DOM element. (Inherited from Control.)
	clientToDoc	Transforms a point from client to document coordinates. (Inherited from CanvasControl.)
S	create	Creates and initializes a new **ItemListView** from the specified element. This method is static and can be called without creating an instance of the class.
	docToClient	Transforms a point from document to client coordinates. (Inherited from CanvasControl.)
S	find	Returns the specified **ItemListView** object. This member is static and can be invoked without creating an instance of the class.
S	fromJson	Constructs a JavaScript object from a JSON string. (Inherited from Control.)
	getItemAt	Gets the item at the specified location. (Inherited from ListViewBase.)
	raiseEvent	Raises an event. (Inherited from Disposable.)
	registerForSubmit	Registers an onsubmit handler for the Control's parent html form to flush postback data. (Inherited from Control.)
	removeEventListener	Removes a single event listener attached to the instance. (Inherited from Disposable.)
	removeItem	Removes an item from the list view. (Inherited from ListViewBase.)
	selectItem	Selects the specified item. (Inherited from ListViewBase.)
S	toJson	Converts a JavaScript object to a JSON string. (Inherited from Control.)

Public Events

	Name	Description
	controlLoaded	Raised when the control is loaded. (Inherited from Control.)
	controlUnloaded	Raised when the control is unloaded. (Inherited from Control.)
	controlUnloading	Raised when the control is unloading. (Inherited from Control.)
	linkSelected	Raised when a link is selected, either programmatically or by the user. (Inherited from ListViewBase.)
	nodeSelected	Raised when a node is selected, either programmatically or by the user. (Inherited from ListViewBase.)
	shapeLibraryLoaded	Raised when a ShapeLibrary is loaded from **shapeLibraryLocation**. (Inherited from ListViewBase.)

See Also

ItemListView Class
MindFusion.Diagramming Namespace

ItemListView Constructor

Creates an instance of ItemListView.

Namespace: MindFusion.Diagramming
Assembly: ItemListView.js

Syntax

JavaScript — Copy Code

```
function ItemListView (element)
```

Parameters

element
HTMLCanvasElement. The Canvas DOM Element this ItemListView is associated with.

See Also

ItemListView Class
MindFusion.Diagramming Namespace

ItemListView Methods

ItemListView.create Method

Creates and initializes a new ItemListView from the specified element. This method is static and can be called without creating an instance of the class.

Namespace: MindFusion.Diagramming
Assembly: ItemListView.js

Syntax

JavaScript — Copy Code

```
function create (element)
```

Parameters

element
HTMLCanvasElement. The DOM element that the ItemListView should be attached to.

Return Value

ItemListView. An **ItemListView** object that represents the newly created view.

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming Namespace

ItemListView.find Method

Returns the specified ItemListView object. This member is static and can be invoked without creating an instance of the class.

Namespace: MindFusion.Diagramming
Assembly: ItemListView.js

Syntax

JavaScript Copy Code

```
function find (id)
```

Parameters

id

String. The id of the ItemListView's DOM element.

Return Value

ItemListView. An ItemListView object with the specified id, if found; otherwise, null.

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming Namespace

ItemListView Properties

ItemListView.autoConnectDistance Property

Gets or sets the maximum distance to a node when auto connecting links.

Namespace: MindFusion.Diagramming
Assembly: ItemListView.js

Syntax

JavaScript Copy Code

```
get autoConnectDistance() {}
set autoConnectDistance(value) {}
```

Property Value

Number

A number specifying auto-connect distance.

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming Namespace

ItemListView.autoConnectLinks Property

Gets or sets whether links should be automatically connected to the nearest node.

Namespace: MindFusion.Diagramming
Assembly: ItemListView.js

Syntax

JavaScript Copy Code

```
get autoConnectLinks() {}
set autoConnectLinks(value) {}
```

Property Value

Boolean

true to automatically connect links, or **false** otherwise. The default value is **true**.

Remarks

If this property is enabled and there are nodes within autoConnectDistance when a link is created via drag-and-drop, the link automatically connects to the nodes. Otherwise it is created as unconnected link.

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming Namespace

ItemListView.columns Property

Gets or sets the number of columns in control's layout.

Namespace: MindFusion.Diagramming
Assembly: ItemListView.js

Syntax

JavaScript Copy Code

```
get columns() {}
set columns(value) {}
```

Property Value

Number

An integer value specifying the number of columns. The default value is 1.

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming Namespace

ItemListView.labelPosition Property

Gets or sets a value indicating the position of labels relative to item icons.

Namespace: MindFusion.Diagramming
Assembly: ItemListView.js

Syntax

JavaScript Copy Code

```
get labelPosition() {}
set labelPosition(value) {}
```

Property Value

LabelPosition

A member of the LabelPosition enumeration. The default value is **Auto**.

Remarks

The default Auto value places labels to the right of item icons if using one-column layout, and below icons if showing multiple columns.

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming Namespace

ItemListView Events

ListViewBase.linkSelected Event

Raised when a link is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript — Copy Code

```
listView.linkSelected.addEventListener(onLinkSelected);
```

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase.nodeSelected Event

Raised when a node is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript — Copy Code

```
listView.nodeSelected.addEventListener(onNodeSelected);
```

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase.shapeLibraryLoaded Event

Raised when a ShapeLibrary is loaded from shapeLibraryLocation.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript — Copy Code

```
listView.shapeLibraryLoaded.addEventListener(onShapeLibraryLoaded);
```

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

LabelPosition Enumeration

Specifies the position of labels in a ListView control.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
LabelPosition = {}
```

Members

	Member name	Description
	Auto	Labels are positioned depending on the column count.
	Right	Labels are on the right side of the icon.
	Left	Labels are on the left side of the icon.
	Bottom	Labels are positioned below the icon.

See Also

MindFusion.Diagramming Namespace

LengthType Enumeration

Indicates how GridPanel determines dimensions of its rows and columns.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
LengthType = {}
```

Members

	Member name	Description
	Auto	The row or column is auto-sized to fit its child components.
	Relative	The size of relative elements is calculated from available space in the GridPanel proportionally to the number of other relative elements.

Remarks

Members of this enumeration can be assigned to the lengthType property of GridRow and GridColumn.

See Also

MindFusion.Diagramming Namespace

LinkCrossingEventArgs Class

Contains the arguments passed to handlers of the validateLinkCrossing event.

Syntax

JavaScript Copy Code

```
// class
LinkCrossingEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.CancelEventArgs
MindFusion.Diagramming.LinkCrossingEventArgs

See Also

LinkCrossingEventArgs Members
MindFusion.Diagramming Namespace

LinkCrossingEventArgs Members

Contains the arguments passed to handlers of the validateLinkCrossing event.

The following tables list the members exposed by the LinkCrossingEventArgs type.

Public Constructors

	Name	Description
	LinkCrossingEventArgs	Initializes a new instance of the LinkCrossingEventArgs class.

Public Properties

	Name	Description
	bottomLink	The link at lower Z position.
	bottomSegmentIndex	Index of bottom link's segment that contains the intersection point.
	intersectionPoint	Intersection point that will show crossing decoration if the event is validated.
	topLink	The link at higher Z position.
	topSegmentIndex	Index of top link's segment that contains the intersection point.

See Also

LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs Constructor

Initializes a new instance of the LinkCrossingEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function LinkCrossingEventArgs (args)
```

Parameters

args

See Also

LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs Properties

LinkCrossingEventArgs.bottomLink Property

The link at lower Z position.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get bottomLink() {}
set bottomLink(value) {}
```

Property Value

DiagramLink

An instance of the DiagramLink class.

See Also

LinkCrossingEventArgs Members
LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs.bottomSegmentIndex Property

Index of bottom link's segment that contains the intersection point.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get bottomSegmentIndex() {}
set bottomSegmentIndex(value) {}
```

Property Value

Number

An integer index of link's segment containing intersectionPoint.

See Also

LinkCrossingEventArgs Members
LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs.intersectionPoint Property

Intersection point that will show crossing decoration if the event is validated.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get intersectionPoint() {}
set intersectionPoint(value) {}
```

Property Value

Point

An instance of the Point class containing intersection coordinates.

See Also

LinkCrossingEventArgs Members
LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs.topLink Property

The link at higher Z position.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get topLink() {}
set topLink(value) {}
```

Property Value

DiagramLink

An instance of the DiagramLink class.

See Also

LinkCrossingEventArgs Members
LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs.topSegmentIndex Property

Index of top link's segment that contains the intersection point.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get topSegmentIndex() {}
set topSegmentIndex(value) {}
```

Property Value

Number

An integer index of link's segment containing intersectionPoint.

See Also

LinkCrossingEventArgs Members
LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossings Enumeration

Defines values that indicate how to render link intersection points.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
LinkCrossings = {}
```

Members

	Member name	Description
	Straight	No special indication for link intersection points.
	Arcs	An arc is drawn at the intersection point of two links. It connects the segments of the link at higher Z position and goes over the link at lower Z.
	Cut	The intersection segments of links at lower Z position are cut by links at higher Z position.

Remarks

Members of this enumeration can be assigned to the linkCrossings property of Diagram.

See Also

MindFusion.Diagramming Namespace

LinkEventArgs Class

Contains the arguments passed to handlers of link-related events.

Syntax

JavaScript Copy Code

```
// class
LinkEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.ItemEventArgs
  MindFusion.Diagramming.LinkEventArgs

See Also

LinkEventArgs Members
MindFusion.Diagramming Namespace
Events Class

LinkEventArgs Members

Contains the arguments passed to handlers of link-related events.

The following tables list the members exposed by the LinkEventArgs type.

Public Constructors

	Name	Description
	LinkEventArgs	Initializes a new instance of the **LinkEventArgs** class.

Public Properties

	Name	Description
	destination	Gets the candidate destination node; reported to validation events.
	Index	Gets an integer index related to the event. (Inherited from ItemEventArgs.)
	Item	Gets the item for which the event is raised. (Inherited from ItemEventArgs.)
	label	Gets the link label for which the event is raised.
	link	Gets the link for which the event is raised.
	origin	Gets the candidate origin node; reported to validation events.

See Also

LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs Constructor

Initializes a new instance of the LinkEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function LinkEventArgs (args)
```

Parameters

args

See Also

LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs Properties

LinkEventArgs.destination Property

Gets the candidate destination node; reported to validation events.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get destination() {}
set destination(value) {}
```

Property Value

DiagramNode

A DiagramNode instance.

Remarks

When handling linkCreating and linkModifying validation events, origin and **destination** values of LinkEventArgs are generally different from link's current origin and destination. If you do not cancel the operation, the former are assigned to the latter once the user stops drawing the link.

See Also

LinkEventArgs Members
LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs.label Property

Gets the link label for which the event is raised.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get label() {}
```

Property Value

LinkLabel

A LinkLabel instance.

See Also

LinkEventArgs Members
LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs.link Property

Gets the link for which the event is raised.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get link() {}
```

Property Value

DiagramLink

A DiagramLink instance.

See Also

LinkEventArgs Members
LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs.origin Property

Gets the candidate origin node; reported to validation events.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript — Copy Code

```
get origin() {}
set origin(value) {}
```

Property Value

DiagramNode

A DiagramNode instance.

Remarks

When handling linkCreating and linkModifying validation events, **origin** and destination values of LinkEventArgs are generally different from link's current origin and destination. If you do not cancel the operation, the former are assigned to the latter once the user stops drawing the link.

See Also

LinkEventArgs Members
LinkEventArgs Class
MindFusion.Diagramming Namespace
linkCreating Event
linkModifying Event

LinkLabel Class

Represents a label displayed by a DiagramLink. New labels can be added to a link by calling its addLabel method.

Syntax

JavaScript — Copy Code

```
// class
LinkLabel.prototype = {}
```

Remarks

The position of a label is determined by where it is pinned along the link's geometry, and offset from the pin point, as set using setControlPointPosition, setSegmentPosition and setLinkLengthPosition methods. In addition, the horizontalAlign and verticalAlign properties specify whether the label is centered on its pin point or aligned to it.

Set the autoArrange property to let the diagram set label's position automatically, avoiding overlaps with nodes and other link labels.

You can let users arrange labels manually by setting the behavior property of DiagramView to MoveLabels.

Inheritance Hierarchy

MindFusion.Diagramming.ItemLabel
  MindFusion.Diagramming.LinkLabel

See Also

LinkLabel Members
MindFusion.Diagramming Namespace

LinkLabel Members

Represents a label displayed by a DiagramLink. New labels can be added to a link by calling its addLabel method.

The following tables list the members exposed by the LinkLabel type.

Public Constructors

	Name	Description
	LinkLabel	Initializes a new instance of the LinkLabel class.

Public Properties

	Name	Description
	autoArrange	Gets or sets whether to automatically arrange the label to avoid overlapping nodes.
	brush	Gets or sets the brush used to draw the label's background.
	Brush	Gets or sets the brush used to draw the label's background. (Inherited from ItemLabel.)
	ContentPadding	Gets or sets the distance between image and text when both are specified. (Inherited from ItemLabel.)
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled. (Inherited from ItemLabel.)
	EnableWrap	Gets or sets a value indicating whether the label's text can be wrapped. (Inherited from ItemLabel.)
	font	Gets or sets the font used to render label's text.
	Font	Gets or sets the label's font. (Inherited from ItemLabel.)
	horizontalAlign	Gets or sets the label's horizontal alignment.
	HorizontalAlign	Gets or sets the label's horizontal alignment. (Inherited from ItemLabel.)
	HorizontalOffset	Gets or sets the label's horizontal offset from its pin point. (Inherited from ItemLabel.)
	Image	Gets or sets the image displayed by this label. (Inherited from ItemLabel.)
	ImageAlign	Gets or sets the alignment of label's image relative to text. (Inherited from ItemLabel.)
	ImageSize	Gets or sets a size that overrides the native image size. (Inherited from ItemLabel.)
	ImageUrl	Gets or sets URL of the image to display inside this label. (Inherited from ItemLabel.)
	Index	Gets or sets the index of an edge or point this label is relative to. (Inherited from ItemLabel.)
	Item	Gets or sets the DiagramItem that displays this label. (Inherited from ItemLabel.)
	Locked	Gets or sets a value indicating whether users are allowed to edit this label. (Inherited from ItemLabel.)
	margin	Gets or sets the label's margin.
	Margin	Gets or sets the label's margin. (Inherited from ItemLabel.)
	MaxWidth	Gets or sets the maximum width of the label. (Inherited from ItemLabel.)
	text	Gets or sets the label's text.
	Text	Gets or sets the label's text. (Inherited from ItemLabel.)
	TextBrush	Gets or sets the brush used to draw the label's text. (Inherited from ItemLabel.)

	textColor	Gets or sets the text color.
	verticalAlign	Gets or sets the label's vertical alignment.
	VerticalAlign	Gets or sets the label's vertical alignment. (Inherited from ItemLabel.)
	VerticalOffset	Gets or sets the label's vertical offset from its pin point. (Inherited from ItemLabel.)

Public Methods

	Name	Description
	Cancel	Called when drag operation is canceled. (Inherited from ItemLabel.)
	Draw	Draws the label on the specified IGraphics surface. (Inherited from ItemLabel.)
	GetEditRect	InplaceEditable.GetEditRect implementation. (Inherited from ItemLabel.)
	GetTextToEdit	InplaceEditable.GetTextToEdit implementation. (Inherited from ItemLabel.)
	LoadFromXml	Loads the label from the specified XML element. (Inherited from ItemLabel.)
	OnMouseDown	Called when the left mouse button is pressed while the pointer is over this label. (Inherited from ItemLabel.)
	OnMouseMove	Called when the mouse is moved while this label tracks the mouse events. (Inherited from ItemLabel.)
	OnMouseUp	Called when the left mouse button is released while this label tracks the mouse events. (Inherited from ItemLabel.)
	SaveToXml	Saves the label into the specified XML element. (Inherited from ItemLabel.)
	setControlPointPosition	Positions this label relatively to the specified control point.
	SetEditedText	InplaceEditable.SetEditedText implementation. (Inherited from ItemLabel.)
	setLinkLengthPosition	Positions this label relatively to the link length.
	SetMouseCursor	Indicates what cursor to display for interactive labels. (Inherited from ItemLabel.)
	setSegmentPosition	Positions this label relatively to the specified link segment.
	UpdateValue	Called while drag operation is in progress. (Inherited from ItemLabel.)

Protected Methods

	Name	Description
	LoadFromJson	Loads the label from a JsonObject. (Inherited from ItemLabel.)
	SaveToJson	Saves the label into a **JsonObject**. (Inherited from ItemLabel.)

See Also

LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel Constructor

Initializes a new instance of the LinkLabel class.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
function LinkLabel (link, text)
```

Parameters

link
DiagramLink. The DiagramLink that displays the label.

text
String. The label's text.

See Also

LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel Methods

LinkLabel.setControlPointPosition Method

Positions this label relatively to the specified control point.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
function setControlPointPosition (controlPoint, dx, dy)
```

Parameters

controlPoint
Number. The control point index.

dx
Number. Horizontal offset from the control point.

dy
Number. Vertical offset from the control point.

Remarks

This method sets the specified control point as label's pin point and assigns specified offsets to horizontalOffset and verticalOffset properties.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.setLinkLengthPosition Method

Positions this label relatively to the link length.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
function setLinkLengthPosition (lengthFactor)
```

Parameters

lengthFactor
Number. A value between 0 and 1 specifying the position of the label relatively to the link's length.

Remarks

This method sets the point at specified distance along link's geometry as pin point and assigns specified offsets to horizontalOffset and verticalOffset properties.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.setSegmentPosition Method

Positions this label relatively to the specified link segment.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
function setSegmentPosition (segment, dx, dy)
```

Parameters

segment
Number. The segment index.

dx
Number. Horizontal offset from the segment center.

dy
Number. Vertical offset from the segment center.

Remarks

This method sets the middle of specified segment as label's pin point and assigns specified offsets to horizontalOffset and verticalOffset properties.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel Properties

LinkLabel.autoArrange Property

Gets or sets whether to automatically arrange the label to avoid overlapping nodes.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
get autoArrange() {}
set autoArrange(value) {}
```

Property Value

Boolean

true if the label should be auto-arranged, or **false** otherwise.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.brush Property

Gets or sets the brush used to draw the label's background.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
get brush() {}
set brush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.font Property

Gets or sets the font used to render label's text.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
get font() {}
set font(value) {}
```

Property Value

Font

An instance of the Font class.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.horizontalAlign Property

Gets or sets the label's horizontal alignment.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
get horizontalAlign() {}
set horizontalAlign(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.margin Property

Gets or sets the label's margin.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
get margin() {}
set margin(value) {}
```

Property Value

Thickness

A Thickness instance specifying the margin space on all sides of label.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.text Property

Gets or sets the label's text.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
get text() {}
set text(value) {}
```

Property Value

String

A string containing the label's text.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.textColor Property

Gets or sets the text color.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

A string value identifying the text color.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.verticalAlign Property

Gets or sets the label's vertical alignment.

Namespace: MindFusion.Diagramming
Assembly: LinkLabel.js

Syntax

JavaScript Copy Code

```
get verticalAlign() {}
set verticalAlign(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkNodesBehavior Class

The base class for behavior types that allow drawing both nodes and links.

Syntax

JavaScript Copy Code

```
// class
LinkNodesBehavior.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.BehaviorBase
MindFusion.Diagramming.LinkNodesBehavior

See Also

LinkNodesBehavior Members
MindFusion.Diagramming Namespace

LinkNodesBehavior Members

The base class for behavior types that allow drawing both nodes and links.

The following tables list the members exposed by the LinkNodesBehavior type.

Public Constructors

	Name	Description
	LinkNodesBehavior	Overloaded.

Public Properties

	Name	Description
	diagram	Gets the target diagram of this behavior object. (Inherited from BehaviorBase.)

Public Methods

	Name	Description
	createController	BehaviorBase.createController override.

See Also

LinkNodesBehavior Class
MindFusion.Diagramming Namespace

LinkNodesBehavior Constructor

Overload List

Name	Description
LinkNodesBehavior ()	Initializes a new instance of the LinkNodesBehavior class.
LinkNodesBehavior (Diagram)	Initializes a new instance of the LinkNodesBehavior class.
LinkNodesBehavior (DiagramView)	Initializes a new instance of the LinkNodesBehavior class.

See Also

LinkNodesBehavior Class
MindFusion.Diagramming Namespace

LinkNodesBehavior Constructor ()

Initializes a new instance of the LinkNodesBehavior class.

Namespace: MindFusion.Diagramming
Assembly: LinkNodesBehavior.js

Syntax

JavaScript Copy Code

```
function LinkNodesBehavior ()
```

See Also

LinkNodesBehavior Constructor Overload List
LinkNodesBehavior Class
MindFusion.Diagramming Namespace

LinkNodesBehavior Constructor (Diagram)

Initializes a new instance of the LinkNodesBehavior class.

Namespace: MindFusion.Diagramming
Assembly: LinkNodesBehavior.js

Syntax

JavaScript Copy Code

```
function LinkNodesBehavior (parent)
```

Parameters

parent
The Diagram this behavior will be associated with.

See Also

LinkNodesBehavior Constructor Overload List
LinkNodesBehavior Class
MindFusion.Diagramming Namespace

LinkNodesBehavior Constructor (DiagramView)

Initializes a new instance of the LinkNodesBehavior class.

Namespace: MindFusion.Diagramming
Assembly: LinkNodesBehavior.js

Syntax

JavaScript Copy Code

```
function LinkNodesBehavior (parent)
```

Parameters

parent
DiagramView. The DiagramView this behavior will be associated with.

See Also

LinkNodesBehavior Constructor Overload List
LinkNodesBehavior Class
MindFusion.Diagramming Namespace

LinkNodesBehavior Methods

LinkNodesBehavior.createController Method

BehaviorBase.createController override.

Namespace: MindFusion.Diagramming
Assembly: LinkNodesBehavior.js

Syntax

JavaScript Copy Code

```
function createController (state)
```

Parameters

state
Object. An object representing current context and input event.

Return Value

SinglePointerController. An instance of SinglePointerController -derived class.

See Also

LinkNodesBehavior Members
LinkNodesBehavior Class
MindFusion.Diagramming Namespace

LinkShape Enumeration

Specifies available styles for the segments of links.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
LinkShape = {}
```

Members

	Member name	Description
	Bezier	The segments of a link are Bézier curves.
	Polyline	The segments of a link are straight lines.
	Cascading	The segments of a link are alternating horizontal and vertical lines, orthogonal to each other.
	Spline	The segments of a link are curves forming an interpolating spline that passes through all control points.

Remarks

Members of this enumeration can be assigned to the shape property of DiagramLink.

See Also

MindFusion.Diagramming Namespace

LinkShapesBehavior Class

This class is used when view's behavior is set to Behavior.LinkShapes.

Syntax

JavaScript Copy Code

```
// class
LinkShapesBehavior.prototype = {}
```

Inheritance Hierarchy

See Also

LinkShapesBehavior Members
MindFusion.Diagramming Namespace

LinkShapesBehavior Members

This class is used when view's behavior is set to Behavior.LinkShapes.

The following tables list the members exposed by the LinkShapesBehavior type.

Public Constructors

	Name	Description
	LinkShapesBehavior	Initializes a new instance of the **LinkShapesBehavior** class.

Public Methods

	Name	Description
	createNode	LinkNodesBehavior.CreateNode override.

See Also

LinkShapesBehavior Class
MindFusion.Diagramming Namespace

LinkShapesBehavior Constructor

Initializes a new instance of the LinkShapesBehavior class.

Namespace: MindFusion.Diagramming
Assembly: LinkShapesBehavior.js

Syntax

JavaScript Copy Code

```
function LinkShapesBehavior (parent)
```

Parameters

parent
The Diagram this behavior will be associated with.

See Also

LinkShapesBehavior Class
MindFusion.Diagramming Namespace

LinkShapesBehavior Methods

LinkShapesBehavior.createNode Method

LinkNodesBehavior.CreateNode override.

Namespace: MindFusion.Diagramming
Assembly: LinkShapesBehavior.js

Syntax

JavaScript — Copy Code

```
function createNode ()
```

Return Value

An instance of the DiagramNode class.

See Also

LinkShapesBehavior Members
LinkShapesBehavior Class
MindFusion.Diagramming Namespace

LinkTextStyle Enumeration

Specifies the position and alignment of links' labels.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
LinkTextStyle = {}
```

Members

	Member name	Description
	Center	The text is displayed horizontally above the middle link segment or control point, depending on whetherthere are an odd or even number of segments.
	Rotate	The text is displayed rotated at the same angle as the segment where the text is placed.A long enough segment from the middle link segments is chosen for that purpose.
	OverLongestSegment	The text follows the longest link segment and is displayed centered at the segment's middle point.
	Follow	The label text starts from the first point of the link and follows the path defined by the link segments. This style is useful for displaying long text.
	MiddleSegment	The text is displayed horizontally above the middle link segment. If there are an even number of segments, the label is drawn over the longer from the two segments adjacent to the middle control point.
	MiddleSegmentRotated	The text is displayed rotated at the same angle as the segment where the text is placed. A long enough segmentfrom the middle link segments is chosen for that purpose.

Remarks

Members of this enumeration can be assigned to the textStyle property of DiagramLink.

See Also

MindFusion.Diagramming Namespace

LinkValidationEventArgs Class

Contains the arguments passed to handlers of link validation events.

Syntax

JavaScript Copy Code

```
// class
LinkValidationEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.CancelEventArgs
  MindFusion.Diagramming.ItemEventArgs
    MindFusion.Diagramming.LinkEventArgs
      MindFusion.Diagramming.LinkValidationEventArgs

See Also

LinkValidationEventArgs Members
MindFusion.Diagramming Namespace

LinkValidationEventArgs Members

Contains the arguments passed to handlers of link validation events.

The following tables list the members exposed by the LinkValidationEventArgs type.

Public Constructors

	Name	Description
	LinkValidationEventArgs	Initializes a new instance of the LinkValidationEventArgs class.

Public Properties

	Name	Description
	AdjustmentHandle	Gets or sets the adjustment handle being moved by the user. (Inherited from LinkEventArgs.)
	anchorIndex	Gets the zero-based index of the anchor point related to the event.
	Label	Gets or sets a LinkLabel related to this event. (Inherited from LinkEventArgs.)
	Link	Gets the link for which the event raised. (Inherited from LinkEventArgs.)
	MouseButton	Gets which mouse button has been pressed. (Inherited from LinkEventArgs.)
	MousePosition	Gets or sets the current mouse pointer position. (Inherited from LinkEventArgs.)
	node	Gets the node related to the event.
	tableRow	Gets the zero-based index of the table row related to the event.

See Also

LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs Constructor

Initializes a new instance of the LinkValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function LinkValidationEventArgs (args)
```

Parameters

args

See Also

LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs Properties

LinkValidationEventArgs.anchorIndex Property

Gets the zero-based index of the anchor point related to the event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get anchorIndex() {}
set anchorIndex(value) {}
```

Property Value

Number

The anchor point index.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs.node Property

Gets the node related to the event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get node() {}
set node(value) {}
```

Property Value

DiagramNode

A DiagramNode instance.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs.tableRow Property

Gets the zero-based index of the table row related to the event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get tableRow() {}
set tableRow(value) {}
```

Property Value

Number

The table row index.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

ListViewBase Class

The base class of ItemListView and NodeListView controls.

Syntax

JavaScript Copy Code

```
// class
ListViewBase.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.CanvasControl
  MindFusion.Diagramming.ListViewBase
    MindFusion.Diagramming.ItemListView
    MindFusion.Diagramming.NodeListView

See Also

ListViewBase Members
MindFusion.Diagramming Namespace

ListViewBase Members

The base class of ItemListView and NodeListView controls.

The following tables list the members exposed by the ListViewBase type.

Public Properties

	Name	Description
	backColor	Gets or sets the background color of the list view.
	draggedItem	Gets the currently dragged item.
	element	Returns a reference to the control's DOM element. (Inherited from Control.)
	enabled	Gets or sets whether mouse events are enabled. (Inherited from Control.)
	iconSize	Gets or sets the size of items displayed in this list view.
	items	Gets the items contained in this list view.
	licenseLocation	Gets or sets the location of the license key file. (Inherited from Control.)
	measureUnit	Gets the unit of measure used for logical coordinates. (Inherited from CanvasControl.)
	newInstanceSize	Gets or sets the default size of items created by drag-and-drop from this list view.
	orientation	Gets or sets the orientation of the view layout.
	padding	Gets or sets the padding space between item's slot bounds and its contents.
	shapeLibraryLocation	Gets the location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path.
	textColor	Gets or sets the color used to paint item captions.
	tooltipDelay	Gets or sets the delay before a tooltip is shown.

Public Methods

	Name	Description
	addEventListener	Registers a single event listener on the instance. (Inherited from Disposable.)
S	addHandlers	Adds an event listener to a DOM element. (Inherited from Control.)
	addItem	Adds an item to the list view.
	clearAll	Removes all items from the list view.
S	clearHandlers	Removes all event listeners from a DOM element. (Inherited from Control.)
	clientToDoc	Transforms a point from client to document coordinates. (Inherited from CanvasControl.)
	docToClient	Transforms a point from document to client coordinates. (Inherited from CanvasControl.)
S	fromJson	Constructs a JavaScript object from a JSON string. (Inherited from Control.)
	getItemAt	Gets the item at the specified location.
	raiseEvent	Raises an event. (Inherited from Disposable.)

	registerForSubmit	Registers an onsubmit handler for the Control's parent html form to flush postback data. (Inherited from Control.)
	removeEventListener	Removes a single event listener attached to the instance. (Inherited from Disposable.)
	removeItem	Removes an item from the list view.
	selectItem	Selects the specified item.
	toJson	Converts a JavaScript object to a JSON string. (Inherited from Control.)

Public Events

	Name	Description
	controlLoaded	Raised when the control is loaded. (Inherited from Control.)
	controlUnloaded	Raised when the control is unloaded. (Inherited from Control.)
	controlUnloading	Raised when the control is unloading. (Inherited from Control.)
	linkSelected	Raised when a link is selected, either programmatically or by the user.
	nodeSelected	Raised when a node is selected, either programmatically or by the user.
	shapeLibraryLoaded	Raised when a ShapeLibrary is loaded from **shapeLibraryLocation**.

See Also

ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase Methods

ListViewBase.addItem Method

Adds an item to the list view.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript Copy Code

```
function addItem (item, [caption, [locked, [inactive]]])
```

Parameters

item
DiagramItem. The item to add.

caption
Optional.
String. The caption to display for the item.

locked
Optional.
Boolean. Indicates whether the item can be dragged.

inactive
Optional.
Boolean. Indicates whether the item can be selected.

See Also

ListViewBase Members
ListViewBase Class

ListViewBase.clearAll Method

Removes all items from the list view.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript Copy Code

```
function clearAll ()
```

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace
addItem Method
removeItem Method

ListViewBase.getItemAt Method

Gets the item at the specified location.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript Copy Code

```
function getItemAt (point)
```

Parameters

point
Point. The location where to look.

Return Value

DiagramItem. The item found.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase.removeItem Method

Removes an item from the list view.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript Copy Code

```
function removeItem (item)
```

Parameters

item
DiagramItem. The item to remove.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace
addItem Method
clearAll Method

ListViewBase.selectItem Method

Selects the specified item.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript Copy Code

```
function selectItem (item)
```

Parameters

item
DiagramItem. The item to select.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase Properties

ListViewBase.backColor Property

Gets or sets the background color of the list view.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript Copy Code

```
get backColor() {}
set backColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase.draggedItem Property

Gets the currently dragged item.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript — Copy Code

```
get draggedItem() {}
set draggedItem(value) {}
```

Property Value

DiagramItem

The dragged item.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase.iconSize Property

Gets or sets the size of items displayed in this list view.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript — Copy Code

```
get iconSize() {}
set iconSize(value) {}
```

Property Value

Size

An instance of the Size class specifying item size.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace
newInstanceSize Property
measureUnit Property

ListViewBase.items Property

Gets the items contained in this list view.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript — Copy Code

```
get items() {}
set items(value) {}
```

Property Value

Array

An array of DiagramItem objects to show in the list view.

Remarks

You must call addItem to add items to the list view, instead of modifying this array directly.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace
addItem Method
removeItem Method
clearAll Method

ListViewBase.newInstanceSize Property

Gets or sets the default size of items created by drag-and-drop from this list view.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript Copy Code

```
get newInstanceSize() {}
set newInstanceSize(value) {}
```

Property Value

Size

An instance of the Size class specifying item size.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace
iconSize Property
measureUnit Property

ListViewBase.orientation Property

Gets or sets the orientation of the view layout.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript Copy Code

```
get orientation() {}
set orientation(value) {}
```

Property Value

Orientation

A member of the Orientation enumeration.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase.padding Property

Gets or sets the padding space between item's slot bounds and its contents.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript — Copy Code

```
get padding() {}
set padding(value) {}
```

Property Value

Number

A number that specifies the padding space size.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase.shapeLibraryLocation Property

Gets the location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript — Copy Code

```
get shapeLibraryLocation() {}
set shapeLibraryLocation(value) {}
```

Property Value

String

URL of a shape library file containing custom shape definitions.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase.textColor Property

Gets or sets the color used to paint item captions.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript — Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase.tooltipDelay Property

Gets or sets the delay before a tooltip is shown.

Namespace: MindFusion.Diagramming
Assembly: ListViewBase.js

Syntax

JavaScript Copy Code

```
get tooltipDelay() {}
set tooltipDelay(value) {}
```

Property Value

Number

The tooltip delay in milliseconds.

See Also

ListViewBase Members
ListViewBase Class
MindFusion.Diagramming Namespace

ListViewBase Events

MagnifyBehavior Class

This class is used when view's behavior is set to Behavior.Magnify.

Syntax

JavaScript Copy Code

```
// class
MagnifyBehavior.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.BehaviorBase
  MindFusion.Diagramming.MagnifyBehavior

See Also

MagnifyBehavior Members
MindFusion.Diagramming Namespace

MagnifyBehavior Members

This class is used when view's behavior is set to Behavior.Magnify.

The following tables list the members exposed by the MagnifyBehavior type.

Public Constructors

	Name	Description
	MagnifyBehavior	Overloaded.

Public Properties

	Name	Description
	diagram	Gets the target diagram of this behavior object. (Inherited from BehaviorBase.)

Public Methods

	Name	Description
	createController	Creates a controller appropriate for current context and pointer position. (Inherited from BehaviorBase.)
	onMouseDown	BehaviorBase.onMouseDown override.
	onMouseMove	BehaviorBase.onMouseMove override.
	onMouseUp	BehaviorBase.onMouseUp override.

See Also

MagnifyBehavior Class
MindFusion.Diagramming Namespace

MagnifyBehavior Constructor

Overload List

Name	Description
MagnifyBehavior (Diagram)	Initializes a new instance of the MagnifyBehavior class.
MagnifyBehavior (DiagramView)	Initializes a new instance of the MagnifyBehavior class.

See Also

MagnifyBehavior Class
MindFusion.Diagramming Namespace

MagnifyBehavior Constructor (Diagram)

Initializes a new instance of the MagnifyBehavior class.

Namespace: MindFusion.Diagramming
Assembly: MagnifyBehavior.js

Syntax

JavaScript Copy Code

```
function MagnifyBehavior (parent)
```

Parameters

parent
The Diagram this behavior will be associated with.

See Also

MagnifyBehavior Constructor Overload List
MagnifyBehavior Class
MindFusion.Diagramming Namespace

MagnifyBehavior Constructor (DiagramView)

Initializes a new instance of the MagnifyBehavior class.

Namespace: MindFusion.Diagramming
Assembly: MagnifyBehavior.js

Syntax

JavaScript Copy Code

```
function MagnifyBehavior (parent)
```

Parameters

parent
DiagramView. The DiagramView this behavior will be associated with.

See Also

MagnifyBehavior Constructor Overload List
MagnifyBehavior Class
MindFusion.Diagramming Namespace

MagnifyBehavior Methods

MagnifyBehavior.onMouseDown Method

BehaviorBase.onMouseDown override.

Namespace: MindFusion.Diagramming
Assembly: MagnifyBehavior.js

Syntax

JavaScript Copy Code

```
function onMouseDown (e)
```

Parameters

e
A **MouseButtonEventArgs** that contains the event data.

Return Value

Boolean. true if interaction starts.

See Also

MagnifyBehavior Members
MagnifyBehavior Class
MindFusion.Diagramming Namespace

MagnifyBehavior.onMouseMove Method

BehaviorBase.onMouseMove override.

Namespace: MindFusion.Diagramming
Assembly: MagnifyBehavior.js

Syntax

JavaScript — Copy Code

```
function onMouseMove (e)
```

Parameters

e
A **MouseEventArgs** that contains the event data.

See Also

MagnifyBehavior Members
MagnifyBehavior Class
MindFusion.Diagramming Namespace

MagnifyBehavior.onMouseUp Method

BehaviorBase.onMouseUp override.

Namespace: MindFusion.Diagramming
Assembly: MagnifyBehavior.js

Syntax

JavaScript — Copy Code

```
function onMouseUp (e)
```

Parameters

e
A **MouseButtonEventArgs** that contains the event data.

See Also

MagnifyBehavior Members
MagnifyBehavior Class
MindFusion.Diagramming Namespace

Manipulator Class

Manipulator is an abstract base class which defines the behavior of miscellaneous UI widgets attached to nodes.

Syntax

JavaScript — Copy Code

```
// class
Manipulator.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.Container
  MindFusion.Diagramming.Manipulator

See Also

Manipulator Members
MindFusion.Diagramming Namespace

Manipulator Members

Manipulator is an abstract base class which defines the behavior of miscellaneous UI widgets attached to nodes.

The following tables list the members exposed by the Manipulator type.

Public Constructors

	Name	Description
	Manipulator	Initializes a new instance of the Manipulator class.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

See Also

Manipulator Class
MindFusion.Diagramming Namespace

Manipulator Constructor

Initializes a new instance of the Manipulator class.

Namespace: MindFusion.Diagramming
Assembly: Manipulator.js

Syntax

JavaScript Copy Code

```
function Manipulator ([node])
```

Parameters

node

Optional.
DiagramNode. The DiagramNode containing this manipulator.

See Also

Manipulator Class
MindFusion.Diagramming Namespace

MarkStyle Enumeration

Specifies the visual style of anchor point marks.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
MarkStyle = {}
```

Members

	Member name	Description
	None	No visual indication for the anchor point.
	Cross	Anchor point marked with a cross.
	X	Anchor point marked with an 'X'.
	Circle	Anchor point marked with a circle.
	Rectangle	Anchor point marked with a rectangle.
	Custom	Raises drawAnchorPoint event, which lets you draw custom graphics using the Canvas API.

Remarks

Members of this enumeration can be assigned to the markStyle property of AnchorPoint.

See Also

MindFusion.Diagramming Namespace

ModificationStart Enumeration

Specifies how the user can start modifying an item.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
ModificationStart = {}
```

Members

	Member name	Description
	SelectedOnly	Only selected objects can be modified.
	AutoHandles	Adjustment handles appear automatically around the object under the mouse; it can be modified without selecting it first.

Remarks

Members of this enumeration can be assigned to the modificationStart property of DiagramView.

See Also

MindFusion.Diagramming Namespace

ModifierKeyAction Enumeration

Defines the actions that can be assigned to a modifier key such as CTRL or ALT.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
ModifierKeyAction = {}
```

Members

	Member name	Description
	None	Specifies that a modifier key does not have any specific function assigned to it.
	Pan	Pan the view if the mouse is dragged while the modifier key is pressed down.
	Select	Start drawing a selection rectangle if the mouse is dragged while a modifier key is pressed down. Toggle the selection if an item is clicked while the key is down.
	OverrideBehavior	While the modifier key is pressed down, dragging the mouse starts creating a new item or drawing a selection rectangle instead of modifying a selected item. Otherwise, dragging the mouse could start modification of the selected item.
	Magnify	Magnifies the diagram objects under the mouse while the modifier key is pressed down.
	Zoom	Start drawing a zoom-lasso if the mouse is dragged while respective modifier key is pressed down.

Remarks

Members of this enumeration can be assigned to the properties of view's modifierKeyActions.

See Also

MindFusion.Diagramming Namespace

ModifierKeyActions Class

Maps keyboard modifier keys to special actions in MindFusion.Diagramming such as panning and drawing selection rectangle.

Syntax

JavaScript — Copy Code

```
// class
ModifierKeyActions.prototype = {}
```

Remarks

The modifierKeyActions property of DiagramView returns an object from this class, letting you specify the modifier key functions for each view instance.

Assign member of the ModifierKeyAction enumeration to the alt, control and shift properties to specify what function is assigned to the ALT, CTRL and SHIFT keys.

Inheritance Hierarchy

MindFusion.Diagramming.ModifierKeyActions

See Also

ModifierKeyActions Members
MindFusion.Diagramming Namespace

ModifierKeyActions Members

Maps keyboard modifier keys to special actions in MindFusion.Diagramming such as panning and drawing selection rectangle.

The following tables list the members exposed by the ModifierKeyActions type.

Public Constructors

	Name	Description
	ModifierKeyActions	Initializes a new instance of the ModifierKeyActions class.

Public Properties

	Name	Description
	alt	Gets or sets the action triggered by the ALT key.
	control	Gets or sets the action triggered by the CTRL key.
	shift	Gets or sets the action triggered by the SHIFT key.

See Also

ModifierKeyActions Class
MindFusion.Diagramming Namespace

ModifierKeyActions Constructor

Initializes a new instance of the ModifierKeyActions class.

Namespace: MindFusion.Diagramming
Assembly: ModifierKeyActions.js

Syntax

JavaScript Copy Code

```
function ModifierKeyActions ()
```

See Also

ModifierKeyActions Class
MindFusion.Diagramming Namespace

ModifierKeyActions Properties

ModifierKeyActions.alt Property

Gets or sets the action triggered by the ALT key.

Namespace: MindFusion.Diagramming
Assembly: ModifierKeyActions.js

Syntax

JavaScript Copy Code

```
get alt() {}
set alt(value) {}
```

Property Value

ModifierKeyAction

A member of the ModifierKeyAction enumeration.

Remarks

The value of this property specifies what happens when a user holds down the ALT key and drags the mouse. The default value Pan makes the control enter pan mode, letting the user scroll the diagram contents by dragging it with the mouse. You can change the function of ALT by assigning a different value to this property.

See Also

ModifierKeyActions Members
ModifierKeyActions Class
MindFusion.Diagramming Namespace

ModifierKeyActions.control Property

Gets or sets the action triggered by the CTRL key.

Namespace: MindFusion.Diagramming
Assembly: ModifierKeyActions.js

Syntax

JavaScript Copy Code

```
get control() {}
set control(value) {}
```

Property Value

ModifierKeyAction

A member of the ModifierKeyAction enumeration.

Remarks

The value of this property specifies what happens when a user holds down the CTRL key and drags the mouse. The default value Select makes the control enter selection mode, letting the user draw a lasso rectangle around diagram items in order to select them. You can change the function of CTRL by assigning a different value to this property.

See Also

ModifierKeyActions Members
ModifierKeyActions Class
MindFusion.Diagramming Namespace

ModifierKeyActions.shift Property

Gets or sets the action triggered by the SHIFT key.

Namespace: MindFusion.Diagramming
Assembly: ModifierKeyActions.js

Syntax

JavaScript Copy Code

```
get shift() {}
set shift(value) {}
```

Property Value

ModifierKeyAction

A member of the ModifierKeyAction enumeration.

Remarks

The value of this property specifies what happens when a user holds down the SHIFT key and drags the mouse. The default value OverrideBehavior makes the control ignore item's selection handles, thus not allowing modifying a selected item but starts the creation of a new item. You can change the function of SHIFT by assigning a different value to this property.

See Also

ModifierKeyActions Members
ModifierKeyActions Class
MindFusion.Diagramming Namespace

ModifyBehavior Class

This class is used when view's behavior is set to Behavior.Modify.

Syntax

JavaScript Copy Code

```
// class
ModifyBehavior.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.BehaviorBase
  MindFusion.Diagramming.ModifyBehavior

See Also

ModifyBehavior Members
MindFusion.Diagramming Namespace

ModifyBehavior Members

This class is used when view's behavior is set to Behavior.Modify.

The following tables list the members exposed by the ModifyBehavior type.

Public Constructors

	Name	Description
	ModifyBehavior	Overloaded.

Public Properties

	Name	Description
	diagram	Gets the target diagram of this behavior object. (Inherited from BehaviorBase.)

Public Methods

	Name	Description
	createController	BehaviorBase.createController override.

See Also

ModifyBehavior Constructor

Overload List

Name	Description
ModifyBehavior (Diagram)	Initializes a new instance of the ModifyBehavior class.
ModifyBehavior (DiagramView)	Initializes a new instance of the ModifyBehavior class.

See Also

ModifyBehavior Class
MindFusion.Diagramming Namespace

ModifyBehavior Constructor (Diagram)

Initializes a new instance of the ModifyBehavior class.

Namespace: MindFusion.Diagramming
Assembly: ModifyBehavior.js

Syntax

JavaScript Copy Code

```
function ModifyBehavior (parent)
```

Parameters

parent
The Diagram this behavior will be associated with.

See Also

ModifyBehavior Constructor Overload List
ModifyBehavior Class
MindFusion.Diagramming Namespace

ModifyBehavior Constructor (DiagramView)

Initializes a new instance of the ModifyBehavior class.

Namespace: MindFusion.Diagramming
Assembly: ModifyBehavior.js

Syntax

JavaScript Copy Code

```
function ModifyBehavior (parent)
```

Parameters

parent
DiagramView. The DiagramView this behavior will be associated with.

See Also

ModifyBehavior Constructor Overload List
ModifyBehavior Class

ModifyBehavior Methods

ModifyBehavior.createController Method

BehaviorBase.createController override.

Namespace: MindFusion.Diagramming
Assembly: ModifyBehavior.js

Syntax

JavaScript Copy Code

```
function createController (state)
```

Parameters

state
Object. An object representing current context and input event.

Return Value

SinglePointerController. An instance of SinglePointerController -derived class.

See Also

ModifyBehavior Members
ModifyBehavior Class
MindFusion.Diagramming Namespace

ModifyLinkController Class

A controller used to modify existing links in the diagram.

Syntax

JavaScript Copy Code

```
// class
ModifyLinkController.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.SinglePointerController
MindFusion.Diagramming.ModifyLinkController

See Also

ModifyLinkController Members
MindFusion.Diagramming Namespace

ModifyLinkController Members

A controller used to modify existing links in the diagram.

The following tables list the members exposed by the ModifyLinkController type.

Public Constructors

	Name	Description

	ModifyLinkController	Initializes a new instance of the ModifyLinkController class.

Public Methods

	Name	Description
	cancel	Called when the user stops drawing or modifying a diagram element, rejecting the operation. (Inherited from SinglePointerController.)
	commit	SinglePointerController.commit override.
	drawInteraction	Called to draw current state of user interaction. (Inherited from SinglePointerController.)
	move	SinglePointerController.move override.
	start	SinglePointerController.start override.
	validate	SinglePointerController.validate override.

See Also

ModifyLinkController Class
MindFusion.Diagramming Namespace

ModifyLinkController Constructor

Initializes a new instance of the ModifyLinkController class.

Namespace: MindFusion.Diagramming
Assembly: ModifyLinkController.js

Syntax

JavaScript Copy Code

```
function ModifyLinkController (link, adjustmentHandle)
```

Parameters

link
DiagramLink. The link that will be modified by this controller.

adjustmentHandle
Object. An object representing the adjustment handle used to modify the link.

See Also

ModifyLinkController Class
MindFusion.Diagramming Namespace

ModifyLinkController Methods

ModifyLinkController.commit Method

SinglePointerController.commit override.

Namespace: MindFusion.Diagramming
Assembly: ModifyLinkController.js

Syntax

JavaScript Copy Code

```
function commit (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

ModifyLinkController Members
ModifyLinkController Class
MindFusion.Diagramming Namespace

ModifyLinkController.move Method

SinglePointerController.move override.

Namespace: MindFusion.Diagramming
Assembly: ModifyLinkController.js

Syntax

JavaScript Copy Code

```
function move (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

ModifyLinkController Members
ModifyLinkController Class
MindFusion.Diagramming Namespace

ModifyLinkController.start Method

SinglePointerController.start override.

Namespace: MindFusion.Diagramming
Assembly: ModifyLinkController.js

Syntax

JavaScript Copy Code

```
function start (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

ModifyLinkController Members
ModifyLinkController Class
MindFusion.Diagramming Namespace

ModifyLinkController.validate Method

SinglePointerController.validate override.

Namespace: MindFusion.Diagramming
Assembly: ModifyLinkController.js

Syntax

JavaScript — Copy Code

```
function validate (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

Return Value

Boolean. true to accept the operation, or false otherwise.

See Also

ModifyLinkController Members
ModifyLinkController Class
MindFusion.Diagramming Namespace

ModifyNodeController Class

A controller used to modify existing nodes in the diagram.

Syntax

JavaScript — Copy Code

```
// class
ModifyNodeController.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.SinglePointerController
  MindFusion.Diagramming.ModifyNodeController

See Also

ModifyNodeController Members
MindFusion.Diagramming Namespace

ModifyNodeController Members

A controller used to modify existing nodes in the diagram.

The following tables list the members exposed by the ModifyNodeController type.

Public Constructors

	Name	Description
	ModifyNodeController	Initializes a new instance of the ModifyNodeController class.

Public Methods

	Name	Description
	cancel	Called when the user stops drawing or modifying a diagram element, rejecting the operation. (Inherited from SinglePointerController.)
	commit	SinglePointerController.commit override.

	drawInteraction	Called to draw current state of user interaction. (Inherited from SinglePointerController.)
	move	SinglePointerController.move override.
	start	SinglePointerController.start override.
	validate	SinglePointerController.validate override.

See Also

ModifyNodeController Class
MindFusion.Diagramming Namespace

ModifyNodeController Constructor

Initializes a new instance of the ModifyNodeController class.

Namespace: MindFusion.Diagramming
Assembly: ModifyNodeController.js

Syntax

JavaScript Copy Code

```
function ModifyNodeController (node, adjustmentHandle)
```

Parameters

node
DiagramNode. The node that will be modified by this controller.

adjustmentHandle
Object. An object representing the adjustment handle used to modify the node.

See Also

ModifyNodeController Class
MindFusion.Diagramming Namespace

ModifyNodeController Methods

ModifyNodeController.commit Method

SinglePointerController.commit override.

Namespace: MindFusion.Diagramming
Assembly: ModifyNodeController.js

Syntax

JavaScript Copy Code

```
function commit (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

ModifyNodeController Members
ModifyNodeController Class
MindFusion.Diagramming Namespace

ModifyNodeController.move Method

SinglePointerController.move override.

Namespace: MindFusion.Diagramming
Assembly: ModifyNodeController.js

Syntax

JavaScript — Copy Code

```
function move (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

ModifyNodeController Members
ModifyNodeController Class
MindFusion.Diagramming Namespace

ModifyNodeController.start Method

SinglePointerController.start override.

Namespace: MindFusion.Diagramming
Assembly: ModifyNodeController.js

Syntax

JavaScript — Copy Code

```
function start (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

ModifyNodeController Members
ModifyNodeController Class
MindFusion.Diagramming Namespace

ModifyNodeController.validate Method

SinglePointerController.validate override.

Namespace: MindFusion.Diagramming
Assembly: ModifyNodeController.js

Syntax

JavaScript — Copy Code

```
function validate (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

Return Value

Boolean. true to accept the operation, or false otherwise.

See Also

ModifyNodeController Members
ModifyNodeController Class
MindFusion.Diagramming Namespace

ModifySelectionController Class

A controller used to modify multiple selected items in the diagram.

Syntax

JavaScript Copy Code

```
// class
ModifySelectionController.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.SinglePointerController
MindFusion.Diagramming.ModifySelectionController

See Also

ModifySelectionController Members
MindFusion.Diagramming Namespace

ModifySelectionController Members

A controller used to modify multiple selected items in the diagram.

The following tables list the members exposed by the ModifySelectionController type.

Public Constructors

	Name	Description
	ModifySelectionController	Initializes a new instance of the ModifySelectionController class.

Public Methods

	Name	Description
	cancel	Called when the user stops drawing or modifying a diagram element, rejecting the operation. (Inherited from SinglePointerController.)
	commit	SinglePointerController.commit override.
	drawInteraction	Called to draw current state of user interaction. (Inherited from SinglePointerController.)
	move	SinglePointerController.move override.
	start	SinglePointerController.start override.
	validate	SinglePointerController.validate override.

See Also

ModifySelectionController Class
MindFusion.Diagramming Namespace

ModifySelectionController Constructor

Initializes a new instance of the ModifySelectionController class.

Namespace: MindFusion.Diagramming
Assembly: ModifySelectionController.js

Syntax

JavaScript Copy Code

```
function ModifySelectionController (selection, adjustmentHandle)
```

Parameters

selection
Selection. The selection that will be modified by this controller.

adjustmentHandle
Object. An object representing the adjustment handle used to modify the selection.

See Also

ModifySelectionController Class
MindFusion.Diagramming Namespace

ModifySelectionController Methods

ModifySelectionController.commit Method

SinglePointerController.commit override.

Namespace: MindFusion.Diagramming
Assembly: ModifySelectionController.js

Syntax

JavaScript Copy Code

```
function commit (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

ModifySelectionController Members
ModifySelectionController Class
MindFusion.Diagramming Namespace

ModifySelectionController.move Method

SinglePointerController.move override.

Namespace: MindFusion.Diagramming
Assembly: ModifySelectionController.js

Syntax

JavaScript Copy Code

```
function move (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

ModifySelectionController Members
ModifySelectionController Class
MindFusion.Diagramming Namespace

ModifySelectionController.start Method

SinglePointerController.start override.

Namespace: MindFusion.Diagramming
Assembly: ModifySelectionController.js

Syntax

JavaScript Copy Code

```
function start (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

ModifySelectionController Members
ModifySelectionController Class
MindFusion.Diagramming Namespace

ModifySelectionController.validate Method

SinglePointerController.validate override.

Namespace: MindFusion.Diagramming
Assembly: ModifySelectionController.js

Syntax

JavaScript Copy Code

```
function validate (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

Return Value

Boolean. true to accept the operation, or false otherwise.

See Also

ModifySelectionController Members
ModifySelectionController Class
MindFusion.Diagramming Namespace

MouseButtonActions Enumeration

Specifies what actions can be associated with the middle and right mouse buttons.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
MouseButtonActions = {}
```

Members

	Member name	Description
	None	Do not associate any action with the button.
	Pan	Pans the view if the mouse is dragged while the button is pressed down.
	Cancel	Cancels the current drawing operation.
	Select	Starts a multiple selection operation.
	Draw	Draw new items or modify existing items.
	Magnify	Magnifies the items under the mouse while the button is pressed down.
	Zoom	Draw a zoom-lasso if the mouse is dragged with specified button pressed down.

Remarks

Members of this enumeration can be assigned to the middleButtonActions and rightButtonActions properties of DiagramView.

See Also

MindFusion.Diagramming Namespace

MouseInputDispatcher Class

Processes mouse events and dispatches them to controller objects appropriate for current context.

Syntax

JavaScript — Copy Code

```
// class
MouseInputDispatcher.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.MouseInputDispatcher

See Also

MouseInputDispatcher Members
MindFusion.Diagramming Namespace

MouseInputDispatcher Members

Processes mouse events and dispatches them to controller objects appropriate for current context.

The following tables list the members exposed by the MouseInputDispatcher type.

Public Constructors

	Name	Description
	MouseInputDispatcher	Overloaded.

Public Methods

	Name	Description
	onMouseDown	Called when the user presses down a mouse button.
	onMouseMove	Called when the user moves the mouse.
	onMouseUp	Called when the user releases a mouse button.

See Also

MouseInputDispatcher Class
MindFusion.Diagramming Namespace

MouseInputDispatcher Constructor

Overload List

Name	Description
MouseInputDispatcher (Diagram)	Initializes a new instance of the MouseInputDispatcher class.
MouseInputDispatcher (DiagramView)	Initializes a new instance of the MouseInputDispatcher class.

See Also

MouseInputDispatcher Class
MindFusion.Diagramming Namespace

MouseInputDispatcher Constructor (Diagram)

Initializes a new instance of the MouseInputDispatcher class.

Namespace: MindFusion.Diagramming
Assembly: MouseInputDispatcher.js

Syntax

JavaScript Copy Code

```
function MouseInputDispatcher (diagram)
```

Parameters

diagram
The Diagram instance that should receive mouse input.

See Also

MouseInputDispatcher Constructor Overload List
MouseInputDispatcher Class
MindFusion.Diagramming Namespace

MouseInputDispatcher Constructor (DiagramView)

Initializes a new instance of the MouseInputDispatcher class.

Namespace: MindFusion.Diagramming
Assembly: MouseInputDispatcher.js

Syntax

JavaScript — Copy Code

```
function MouseInputDispatcher (diagramView)
```

Parameters

diagramView
DiagramView. The DiagramView instance that should receive mouse input.

See Also

MouseInputDispatcher Constructor Overload List
MouseInputDispatcher Class
MindFusion.Diagramming Namespace

MouseInputDispatcher Methods

MouseInputDispatcher.onMouseDown Method

Called when the user presses down a mouse button.

Namespace: MindFusion.Diagramming
Assembly: MouseInputDispatcher.js

Syntax

JavaScript — Copy Code

```
function onMouseDown (position, button)
```

Parameters

position
Point. A point specified in diagram coordinates.

button
Number. 0 for left button, 1 for middle button, 2 for right button.

See Also

MouseInputDispatcher Members
MouseInputDispatcher Class
MindFusion.Diagramming Namespace

MouseInputDispatcher.onMouseMove Method

Called when the user moves the mouse.

Namespace: MindFusion.Diagramming
Assembly: MouseInputDispatcher.js

Syntax

JavaScript — Copy Code

```
function onMouseMove (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

MouseInputDispatcher Members
MouseInputDispatcher Class
MindFusion.Diagramming Namespace

MouseInputDispatcher.onMouseUp Method

Called when the user releases a mouse button.

Namespace: MindFusion.Diagramming
Assembly: MouseInputDispatcher.js

Syntax

JavaScript Copy Code

```
function onMouseUp (position, button)
```

Parameters

position
Point. A point specified in diagram coordinates.

button
Number. 0 for left button, 1 for middle button, 2 for right button.

See Also

MouseInputDispatcher Members
MouseInputDispatcher Class
MindFusion.Diagramming Namespace

MouseWheelAction Enumeration

Specifies the type of action performed when the user rotates the mouse wheel.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
MouseWheelAction = {}
```

Members

	Member name	Description
	None	The control does not handle wheel rotation internally, but raises mouseWheel event to allow custom handling.
	Scroll	Indicates that the mouse wheel should scroll the view.
	Zoom	Indicates that the mouse wheel should zoom the view.

Remarks

Members of this enumeration can be assigned to the mouseWheelAction property of DiagramView.

See Also

MindFusion.Diagramming Namespace

MoveNodesBehavior Class

This class is used when view's behavior is set to Behavior.MoveNodes.

Syntax

JavaScript — Copy Code

```
// class
MoveNodesBehavior.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.BehaviorBase
  MindFusion.Diagramming.MoveNodesBehavior

See Also

MoveNodesBehavior Members
MindFusion.Diagramming Namespace

MoveNodesBehavior Members

This class is used when view's behavior is set to Behavior.MoveNodes.

The following tables list the members exposed by the MoveNodesBehavior type.

Public Constructors

	Name	Description
	MoveNodesBehavior	Initializes a new instance of the MoveNodesBehavior class.

Public Properties

	Name	Description
	diagram	Gets the target diagram of this behavior object. (Inherited from BehaviorBase.)

Public Methods

	Name	Description
	createController	BehaviorBase.createController override.

See Also

MoveNodesBehavior Class
MindFusion.Diagramming Namespace

MoveNodesBehavior Constructor

Initializes a new instance of the MoveNodesBehavior class.

Namespace: MindFusion.Diagramming
Assembly: MoveNodesBehavior.js

Syntax

JavaScript — Copy Code

```
function MoveNodesBehavior (parent)
```

Parameters

parent
DiagramView. The DiagramView this behavior will be associated with.

See Also

MoveNodesBehavior Class
MindFusion.Diagramming Namespace

MoveNodesBehavior Methods

MoveNodesBehavior.createController Method

BehaviorBase.createController override.

Namespace: MindFusion.Diagramming
Assembly: MoveNodesBehavior.js

Syntax

JavaScript Copy Code

```
function createController (state)
```

Parameters

state
Object. An object representing current context and input event.

Return Value

MindFusion.Diagramming.SinglePointerController. An instance of SinglePointerController -derived class.

See Also

MoveNodesBehavior Members
MoveNodesBehavior Class
MindFusion.Diagramming Namespace

NodeConstraints Class

Defines specific constraints that can be applied to the size and location of nodes.

Syntax

JavaScript Copy Code

```
// class
NodeConstraints.prototype = {}
```

Remarks

Instances of this class can be assigned to the constraints property of nodes.

Inheritance Hierarchy

MindFusion.Diagramming.NodeConstraints

See Also

NodeConstraints Members
MindFusion.Diagramming Namespace

NodeConstraints Members

Defines specific constraints that can be applied to the size and location of nodes.

The following tables list the members exposed by the NodeConstraints type.

Public Constructors

	Name	Description
	NodeConstraints	Initializes a new instance of the **NodeConstraints** class.

Public Properties

	Name	Description
	keepRatio	Gets or sets a value indicating whether the initial width/height ratio of a node should be preserved.
	maxHeight	Gets or sets the maximal node height allowed.
	maxWidth	Gets or sets the maximal node width allowed.
	minHeight	Gets or sets the minimal node height allowed.
	minWidth	Gets or sets the minimal node width allowed.
	moveDirection	Gets or sets constraints on movement direction.

See Also

NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints Constructor

Initializes a new instance of the NodeConstraints class.

Namespace: MindFusion.Diagramming
Assembly: NodeConstraints.js

Syntax

JavaScript Copy Code

```
function NodeConstraints ()
```

See Also

NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints Properties

NodeConstraints.keepRatio Property

Gets or sets a value indicating whether the initial width/height ratio of a node should be preserved.

Namespace: MindFusion.Diagramming
Assembly: NodeConstraints.js

Syntax

JavaScript Copy Code

```
get keepRatio() {}
set keepRatio(value) {}
```

Property Value

Boolean

true to preserve the initial width/height ratio of a node, or **false** othwerise. The default is **false**.

Remarks

If you enable this property from a custom node class' constructor or an initialization-event handler, the node will keep an aspect ratio of 1:1. Thus you could let users draw perfect square or circle shapes by setting **keepRatio** to **true** from the initializeNode handler.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.maxHeight Property

Gets or sets the maximal node height allowed.

Namespace: MindFusion.Diagramming
Assembly: NodeConstraints.js

Syntax

JavaScript Copy Code

```
get maxHeight() {}
set maxHeight(value) {}
```

Property Value

Number

A number specifying the maximal node height.

Remarks

0 = no height constraints; any other value specifies the maximal node height allowed.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.maxWidth Property

Gets or sets the maximal node width allowed.

Namespace: MindFusion.Diagramming
Assembly: NodeConstraints.js

Syntax

JavaScript Copy Code

```
get maxWidth() {}
set maxWidth(value) {}
```

Property Value

Number

A number specifying the maximal node width.

Remarks

0 = no width constraints; any other value specifies the maximal node width allowed.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.minHeight Property

Gets or sets the minimal node height allowed.

Namespace: MindFusion.Diagramming
Assembly: NodeConstraints.js

Syntax

JavaScript Copy Code

```
get minHeight() {}
set minHeight(value) {}
```

Property Value

Number

A number specifying the minimal node height.

Remarks

0 = no height constraints; any other value specifies the minimal node height allowed.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.minWidth Property

Gets or sets the minimal node width allowed.

Namespace: MindFusion.Diagramming
Assembly: NodeConstraints.js

Syntax

JavaScript Copy Code

```
get minWidth() {}
set minWidth(value) {}
```

Property Value

Number

A number specifying the minimal node width.

Remarks

0 = no width constraints; any other value specifies the minimal node width allowed.

See Also

NodeConstraints Members
NodeConstraints Class

NodeConstraints.moveDirection Property

Gets or sets constraints on movement direction.

Namespace: MindFusion.Diagramming
Assembly: NodeConstraints.js

Syntax

JavaScript — Copy Code

```
get moveDirection() {}
set moveDirection(value) {}
```

Property Value

DirectionConstraint

A member of the DirectionConstraint enumeration. The default value is **None**.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeEffect Class

Represents the base class of all visual effects that can be applied to the nodes in a Diagram.

Syntax

JavaScript — Copy Code

```
// class
NodeEffect.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.NodeEffect

See Also

NodeEffect Members
MindFusion.Diagramming Namespace

NodeEffect Members

Represents the base class of all visual effects that can be applied to the nodes in a Diagram.

The following tables list the members exposed by the NodeEffect type.

See Also

NodeEffect Class
MindFusion.Diagramming Namespace

NodeEventArgs Class

Contains the arguments passed to handlers of node-related events.

Syntax

JavaScript — Copy Code

```
// class
NodeEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.ItemEventArgs
  MindFusion.Diagramming.NodeEventArgs

See Also

NodeEventArgs Members
MindFusion.Diagramming Namespace
Events Class

NodeEventArgs Members

Contains the arguments passed to handlers of node-related events.

The following tables list the members exposed by the NodeEventArgs type.

Public Constructors

	Name	Description
	NodeEventArgs	Initializes a new instance of the **NodeEventArgs** class.

Public Properties

	Name	Description
	container	Gets the container a node has been added to or removed from.
	hyperlink	Gets the hyperlink that was clicked inside a node.
	Index	Gets an integer index related to the event. (Inherited from ItemEventArgs.)
	Item	Gets the item for which the event is raised. (Inherited from ItemEventArgs.)
	label	Gets the node label for which the event is raised.
	node	Gets the node for which the event is raised.

See Also

NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs Constructor

Initializes a new instance of the NodeEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript — Copy Code

```
function NodeEventArgs (args)
```

Parameters

args

See Also

NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs Properties

NodeEventArgs.container Property

Gets the container a node has been added to or removed from.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get container() {}
```

Property Value

ContainerNode

A ContainerNode instance.

See Also

NodeEventArgs Members
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs.hyperlink Property

Gets the hyperlink that was clicked inside a node.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get hyperlink() {}
```

Property Value

String

A string containing clicked hyperlink.

Remarks

This value is valid when handling hyperlinkClicked event.

See Also

NodeEventArgs Members
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs.label Property

Gets the node label for which the event is raised.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript — Copy Code

```
get label() {}
set label(value) {}
```

Property Value

NodeLabel

A NodeLabel instance.

See Also

NodeEventArgs Members
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs.node Property

Gets the node for which the event is raised.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript — Copy Code

```
get node() {}
```

Property Value

DiagramNode

An instance of the DiagramNode class.

See Also

NodeEventArgs Members
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeLabel Class

Represents a supplementary label displayed by a DiagramNode. New labels can be added to a node by calling its addLabel method.

Syntax

JavaScript — Copy Code

```
// class
NodeLabel.prototype = {}
```

Remarks

The position of a label is determined by where it is pinned relative to node's bounding box, and offset from the pin point, as set using setCornerPosition, setEdgePosition and setCenterPosition methods. In addition, the horizontalAlign and verticalAlign properties specify whether the label is centered on its pin point or aligned to it.

You can let users arrange labels manually by setting the behavior property of DiagramView to MoveLabels.

Inheritance Hierarchy

MindFusion.Diagramming.ItemLabel
MindFusion.Diagramming.NodeLabel

See Also

NodeLabel Members
MindFusion.Diagramming Namespace

NodeLabel Members

Represents a supplementary label displayed by a DiagramNode. New labels can be added to a node by calling its addLabel method.

The following tables list the members exposed by the NodeLabel type.

Public Constructors

	Name	Description
	NodeLabel	Initializes a new instance of the **NodeLabel** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to draw the label's background. (Inherited from ItemLabel.)
	ContentPadding	Gets or sets the distance between image and text when both are specified. (Inherited from ItemLabel.)
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled. (Inherited from ItemLabel.)
	EnableWrap	Gets or sets a value indicating whether the label's text can be wrapped. (Inherited from ItemLabel.)
	Font	Gets or sets the label's font. (Inherited from ItemLabel.)
	HorizontalAlign	Gets or sets the label's horizontal alignment. (Inherited from ItemLabel.)
	HorizontalOffset	Gets or sets the label's horizontal offset from its pin point. (Inherited from ItemLabel.)
	Image	Gets or sets the image displayed by this label. (Inherited from ItemLabel.)
	ImageAlign	Gets or sets the alignment of label's image relative to text. (Inherited from ItemLabel.)
	ImageSize	Gets or sets a size that overrides the native image size. (Inherited from ItemLabel.)
	ImageUrl	Gets or sets URL of the image to display inside this label. (Inherited from ItemLabel.)
	Index	Gets or sets the index of an edge or point this label is relative to. (Inherited from ItemLabel.)
	Item	Gets or sets the DiagramItem that displays this label. (Inherited from ItemLabel.)
	Locked	Gets or sets a value indicating whether users are allowed to edit this label. (Inherited from ItemLabel.)
	Margin	Gets or sets the label's margin. (Inherited from ItemLabel.)
	MaxWidth	Gets or sets the maximum width of the label. (Inherited from ItemLabel.)
	Text	Gets or sets the label's text. (Inherited from ItemLabel.)

	TextBrush	Gets or sets the brush used to draw the label's text. (Inherited from ItemLabel.)
	VerticalAlign	Gets or sets the label's vertical alignment. (Inherited from ItemLabel.)
	VerticalOffset	Gets or sets the label's vertical offset from its pin point. (Inherited from ItemLabel.)

Public Methods

	Name	Description
	Cancel	Called when drag operation is canceled. (Inherited from ItemLabel.)
	Draw	Draws the label on the specified IGraphics surface. (Inherited from ItemLabel.)
	GetEditRect	InplaceEditable.GetEditRect implementation. (Inherited from ItemLabel.)
	GetTextToEdit	InplaceEditable.GetTextToEdit implementation. (Inherited from ItemLabel.)
	LoadFromXml	Loads the label from the specified XML element. (Inherited from ItemLabel.)
	OnMouseDown	Called when the left mouse button is pressed while the pointer is over this label. (Inherited from ItemLabel.)
	OnMouseMove	Called when the mouse is moved while this label tracks the mouse events. (Inherited from ItemLabel.)
	OnMouseUp	Called when the left mouse button is released while this label tracks the mouse events. (Inherited from ItemLabel.)
	SaveToXml	Saves the label into the specified XML element. (Inherited from ItemLabel.)
	setCenterPosition	Positions this label relatively to the node center.
	setCornerPosition	Positions this label relatively to the specified corner point.
	setEdgePosition	Positions this label relatively to the specified node edge.
	SetEditedText	InplaceEditable.SetEditedText implementation. (Inherited from ItemLabel.)
	SetMouseCursor	Indicates what cursor to display for interactive labels. (Inherited from ItemLabel.)
	UpdateValue	Called while drag operation is in progress. (Inherited from ItemLabel.)

Protected Methods

	Name	Description
	LoadFromJson	Loads the label from a JsonObject. (Inherited from ItemLabel.)
	SaveToJson	Saves the label into a **JsonObject**. (Inherited from ItemLabel.)

See Also

NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel Constructor

Initializes a new instance of the NodeLabel class.

Namespace: MindFusion.Diagramming
Assembly: NodeLabel.js

Syntax

JavaScript Copy Code

```
function NodeLabel (node, text)
```

Parameters

node
DiagramNode. The DiagramNode that displays the label.

text
String. The label's text.

See Also

NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel Methods

NodeLabel.setCenterPosition Method

Positions this label relatively to the node center.

Namespace: MindFusion.Diagramming
Assembly: NodeLabel.js

Syntax

JavaScript Copy Code

```
function setCenterPosition (dx, dy)
```

Parameters

dx
Number. Horizontal offset from the center point.

dy
Number. Vertical offset from the center point.

Remarks

This method sets the node's center as label's pin point and assigns specified offsets to horizontalOffset and verticalOffset properties.

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel.setCornerPosition Method

Positions this label relatively to the specified corner point.

Namespace: MindFusion.Diagramming
Assembly: NodeLabel.js

Syntax

JavaScript Copy Code

```
function setCornerPosition (cornerPoint, dx, dy)
```

Parameters

cornerPoint
Number. The corner point index.

dx
Number. Horizontal offset from the corner point.

dy
Number. Vertical offset from the corner point.

Remarks

This method sets node's corner as label's pin point and assigns specified offsets to horizontalOffset and verticalOffset properties. The *cornerPoint* index starts counting from top-left corner in clockwise direction:

- 0 desigates top-left corner;
- 1 desigates top-right corner;
- 2 desigates bottom-right corner;
- 3 desigates bottom-left corner;

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel.setEdgePosition Method

Positions this label relatively to the specified node edge.

Namespace: MindFusion.Diagramming
Assembly: NodeLabel.js

Syntax

JavaScript Copy Code

```
function setEdgePosition (edge, dx, dy)
```

Parameters

edge
Number. The edge index.

dx
Number. Horizontal offset from the edge center.

dy
Number. Vertical offset from the edge center.

Remarks

This method sets the middle point of node's edge as label's pin point and assigns specified offsets to horizontalOffset and verticalOffset properties. The *edge* index starts counting from top side in clockwise direction:

- 0 desigates center of top edge;
- 1 desigates center of right edge;
- 2 desigates center of bottom edge;
- 3 desigates center of left edge;

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeListView Class

A list view control that hosts DiagramNodes and supports drag-and-drop to DiagramView.

Syntax

JavaScript Copy Code

```
// class
NodeListView.prototype = {}
```

Remarks

Obsolete

This component has been superseded by ItemListView, which can also display DiagramLink objects and supports multi-column layouts.

Call the addNode method to add a node to the view.

Drag-and-drop operations create a clone of the dragged node by calling its clone method. You can override clone to copy property values of custom node subclasses.

Inheritance Hierarchy

MindFusion.Controls.CanvasControl
  MindFusion.Diagramming.ListViewBase
    MindFusion.Diagramming.NodeListView

See Also

NodeListView Members
MindFusion.Diagramming Namespace
Palette Class

NodeListView Members

A list view control that hosts DiagramNodes and supports drag-and-drop to DiagramView.

The following tables list the members exposed by the NodeListView type.

Public Constructors

	Name	Description
	NodeListView	Creates an instance of NodeListView

Public Properties

	Name	Description
	backColor	Gets or sets the background color of the control.
	defaultNodeSize	Gets or sets the default size of nodes created after dragging from this list view.
	draggedItem	Gets the currently dragged item. (Inherited from ListViewBase.)
	draggedNode	Gets the currently dragged node from this list view.
	element	Returns a reference to the control's DOM element. (Inherited from Control.)
	enabled	Gets or sets whether mouse events are enabled. (Inherited from Control.)
	iconSize	Gets or sets the size of nodes displayed in this list view.
	items	Gets the items contained in this list view. (Inherited from ListViewBase.)
	licenseLocation	Gets or sets the location of the license key file. (Inherited from Control.)
	measureUnit	Gets the unit of measure used for logical coordinates. (Inherited from CanvasControl.)

	newInstanceSize	Gets or sets the default size of items created by drag-and-drop from this list view. (Inherited from ListViewBase.)
	nodes	Gets the nodes contained in this NodeListView.
	orientation	Gets or sets the orientation of the view layout.
	padding	Gets or sets the padding space between the NodeListView control and its contents.
	shapeLibraryLocation	Gets the location of a shape library file containing custom shape definitions.It can be set to a full URL of the file, or a relative file path.
	textColor	Gets or sets the color used to paint text in the NodeListView control.
	tooltipDelay	Gets or sets the delay before a tooltip is shown.

Public Methods

	Name	Description
	addEventListener	Registers a single event listener on the instance. (Inherited from Disposable.)
S	addHandlers	Adds an event listener to a DOM element. (Inherited from Control.)
	addItem	Adds an item to the list view. (Inherited from ListViewBase.)
	addNode	Adds a node to the NodeListView.
	clearAll	Removes all items from the **NodeListView**.
S	clearHandlers	Removes all event listeners from a DOM element. (Inherited from Control.)
	clientToDoc	Transforms a point from client to document coordinates. (Inherited from CanvasControl.)
S	create	Creates and initializes a new **NodeListView** from the specified element. This method is static and can be called without creating an instance of the class.
	docToClient	Transforms a point from document to client coordinates. (Inherited from CanvasControl.)
S	find	Returns the specified **NodeListView** object. This member is static and can be invoked without creating an instance of the class.
S	fromJson	Constructs a JavaScript object from a JSON string. (Inherited from Control.)
	getItemAt	Gets the item at the specified location. (Inherited from ListViewBase.)
	getNodeAt	Gets the item at the specified location.
	raiseEvent	Raises an event. (Inherited from Disposable.)
	registerForSubmit	Registers an onsubmit handler for the Control's parent html form to flush postback data. (Inherited from Control.)
	removeEventListener	Removes a single event listener attached to the instance. (Inherited from Disposable.)
	removeItem	Removes an item from the list view. (Inherited from ListViewBase.)
	removeNode	Removes a node from the list view.
	selectItem	Selects the specified item.
	selectItem	Selects the specified item. (Inherited from ListViewBase.)
S	toJson	Converts a JavaScript object to a JSON string. (Inherited from Control.)

Public Events

	Name	Description
	controlLoaded	Raised when the control is loaded. (Inherited from Control.)
	controlUnloaded	Raised when the control is unloaded. (Inherited from Control.)
	controlUnloading	Raised when the control is unloading. (Inherited from Control.)
	linkSelected	Raised when a link is selected, either programmatically or by the user. (Inherited from ListViewBase.)
	nodeSelected	Raised when a node is selected, either programmatically or by the user.
	shapeLibraryLoaded	Raised when a ShapeLibrary is loaded from **shapeLibraryLocation**.

See Also

NodeListView Class
MindFusion.Diagramming Namespace

NodeListView Constructor

Creates an instance of NodeListView

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript Copy Code

```
function NodeListView (element)
```

Parameters

element
HTMLCanvasElement. The Canvas DOM Element this NodeListView is associated with.

See Also

NodeListView Class
MindFusion.Diagramming Namespace

NodeListView Methods

NodeListView.addNode Method

Adds a node to the NodeListView.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript Copy Code

```
function addNode (node, [caption])
```

Parameters

node
DiagramNode. The node to add.

caption

Optional.
String. The caption to display for the node.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace
nodes Property
removeNode Method
clearAll Method

NodeListView.clearAll Method

Removes all items from the NodeListView.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript Copy Code

```
function clearAll ()
```

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace
addNode Method
removeNode Method

NodeListView.create Method

Creates and initializes a new NodeListView from the specified element. This method is static and can be called without creating an instance of the class.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript Copy Code

```
function create (element)
```

Parameters

element
HTMLCanvasElement. The DOM element that the NodeListView should be attached to.

Return Value

A NodeListView object that represents the newly created list view.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.find Method

Returns the specified NodeListView object. This member is static and can be invoked without creating an instance of the class.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript — Copy Code

```
function find (id)
```

Parameters

id
String. The id of the NodeListView's element.

Return Value

A NodeListView object that contains the list view requested by id, if found, or **null** otherwise.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.getNodeAt Method

Gets the item at the specified location.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript — Copy Code

```
function getNodeAt (point)
```

Parameters

point
Point. The location where to look.

Return Value

DiagramNode. The item found.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.removeNode Method

Removes a node from the list view.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript — Copy Code

```
function removeNode (node)
```

Parameters

node
DiagramNode. The node to remove.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace
addNode Method
clearAll Method

NodeListView.selectItem Method

Selects the specified item.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript Copy Code

```
function selectItem (item)
```

Parameters

item
DiagramNode. The item to select.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView Properties

NodeListView.backColor Property

Gets or sets the background color of the control.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript Copy Code

```
get backColor() {}
set backColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.defaultNodeSize Property

Gets or sets the default size of nodes created after dragging from this list view.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

```
JavaScript                                                    Copy Code

get defaultNodeSize() {}
set defaultNodeSize(value) {}
```

Property Value

Size

An instance of the Size class specifying node size.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.draggedNode Property

Gets the currently dragged node from this list view.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

```
JavaScript                                                    Copy Code

get draggedNode() {}
```

Property Value

DiagramNode

The dragged diagram node.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.iconSize Property

Gets or sets the size of nodes displayed in this list view.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

```
JavaScript                                                    Copy Code

get iconSize() {}
```

Property Value

Size

An instance of the Size class specifying node size.

See Also

NodeListView.nodes Property

Gets the nodes contained in this NodeListView.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript Copy Code

```
get nodes() {}
```

Property Value

Array

An array of DiagramNode objects to show in the list view.

Remarks

You must call addNode to add nodes to the list view, instead of modifying this array directly.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.orientation Property

Gets or sets the orientation of the view layout.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript Copy Code

```
get orientation() {}
set orientation(value) {}
```

Property Value

Orientation

A member of the Orientation enumeration.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.padding Property

Gets or sets the padding space between the NodeListView control and its contents.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript Copy Code

```
get padding() {}
set padding(value) {}
```

Property Value

Number

A number that specifies the padding space size.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.shapeLibraryLocation Property

Gets the location of a shape library file containing custom shape definitions.It can be set to a full URL of the file, or a relative file path.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript Copy Code

```
get shapeLibraryLocation() {}
set shapeLibraryLocation(value) {}
```

Property Value

String

URL of a shape library file containing custom shape definitions.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.textColor Property

Gets or sets the color used to paint text in the NodeListView control.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.tooltipDelay Property

Gets or sets the delay before a tooltip is shown.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript — Copy Code

```
get tooltipDelay() {}
set tooltipDelay(value) {}
```

Property Value

Number

The tooltip delay in milliseconds.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView Events

NodeListView.nodeSelected Event

Raised when a node is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript — Copy Code

```
nodeList.nodeSelected.addEventListener(onNodeSelected);
```

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

NodeListView.shapeLibraryLoaded Event

Raised when a ShapeLibrary is loaded from shapeLibraryLocation.

Namespace: MindFusion.Diagramming
Assembly: NodeListView.js

Syntax

JavaScript — Copy Code

```
nodeList.shapeLibraryLoaded.addEventListener(onShapeLibraryLoaded);
```

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming Namespace

Orientation Enumeration

Specifies the orientation of user interface elements.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
Orientation = {}
```

Members

	Member name	Description
	Auto	Select orientation automatically.
	Horizontal	Specifies horizontal orientation.
	Vertical	Specifies vertical orientation.

See Also

MindFusion.Diagramming Namespace

Overview Class

The **Overview** control provides a scaled-down view of Diagram contents and allows scrolling and zooming a DiagramView.

Syntax

JavaScript — Copy Code

```
// class
Overview.prototype = {}
```

Remarks

To specify which DiagramView instance is controlled by the overview, set the diagramView property. Set the scaleMode property to specify if overview should zoom-to-fit the whole diagram or display it at a fixed scale.

Inheritance Hierarchy

MindFusion.Controls.CanvasControl
  MindFusion.Diagramming.Overview

See Also

Overview Members
MindFusion.Diagramming Namespace

Overview Members

The Overview control provides a scaled-down view of Diagram contents and allows scrolling and zooming a DiagramView.

The following tables list the members exposed by the **Overview** type.

Public Constructors

	Name	Description

	Overview	Initializes a new instance of the Overview class.

Public Properties

	Name	Description
	allowZoom	Gets or sets a value indicating whether users are allowed to zoom the diagramby resizing the overview's viewport tracker rectangle.
	autoScrollAmount	Gets or sets the amount by which to auto-scroll the view when the mouse leaves the view boundaries while drawing.
	backColor	Gets or sets the background color of the overview control.
	diagramView	Gets or sets the **DiagramView** shown by this **Overview**.
	dimColor	Gets or sets a color used to blend with the part of the diagram that is not visible in target DiagramView.
	element	Returns a reference to the control's DOM element. (Inherited from Control.)
	enabled	Gets or sets whether mouse events are enabled. (Inherited from Control.)
	licenseLocation	Gets or sets the location of the license key file. (Inherited from Control.)
	measureUnit	Gets the unit of measure used for logical coordinates. (Inherited from CanvasControl.)
	scaleFactor	Gets or sets a fixed scale factor to apply when FitAll is disabled.
	scaleMode	Gets or sets a value indicating whether to fit the whole diagram inside the overview control.
	scrollX	Gets the overview's horizontal scroll position.
	scrollY	Gets the overview's vertical scroll position.
	viewport	Returns the visible portion of the diagram area.

Public Methods

	Name	Description
	addEventListener	Registers a single event listener on the instance. (Inherited from Disposable.)
S	addHandlers	Adds an event listener to a DOM element. (Inherited from Control.)
S	clearHandlers	Removes all event listeners from a DOM element. (Inherited from Control.)
	clientToDoc	Transforms a point from client to document coordinates. (Inherited from CanvasControl.)
S	create	Creates and initializes a new **Overview** from the specified element. This method is static and can be called without creating an instance of the class.
	docToClient	Transforms a point from document to client coordinates. (Inherited from CanvasControl.)
S	find	Returns the specified **Overview** object. This member is static and can be invoked without creating an instance of the class.
S	fromJson	Constructs a JavaScript object from a JSON string. (Inherited from Control.)
	raiseEvent	Raises an event. (Inherited from Disposable.)
	registerForSubmit	Registers an onsubmit handler for the Control's parent html form to flush postback data. (Inherited from Control.)

	removeEventListener	Removes a single event listener attached to the instance. (Inherited from Disposable.)
	toJson	Converts a JavaScript object to a JSON string. (Inherited from Control.)

Public Events

	Name	Description
	controlLoaded	Raised when the control is loaded. (Inherited from Control.)
	controlUnloaded	Raised when the control is unloaded. (Inherited from Control.)
	controlUnloading	Raised when the control is unloading. (Inherited from Control.)

See Also

Overview Class
MindFusion.Diagramming Namespace

Overview Constructor

Initializes a new instance of the Overview class.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
function Overview (element)
```

Parameters

element
HTMLCanvasElement. The Canvas DOM Element this Overview is associated with.

See Also

Overview Class
MindFusion.Diagramming Namespace

Overview Methods

Overview.create Method

Creates and initializes a new Overview from the specified element. This method is static and can be called without creating an instance of the class.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
function create (element)
```

Parameters

element
HTMLCanvasElement. The DOM element that the overview should be attached to.

Return Value

An Overview object that represents the newly created overview.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview.find Method

Returns the specified Overview object. This member is static and can be invoked without creating an instance of the class.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
function find (id)
```

Parameters

id
String. The id of the Overview's element.

Return Value

An **Overview** object with the specified id, if found, or **null** otherwise.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview Properties

Overview.allowZoom Property

Gets or sets a value indicating whether users are allowed to zoom the diagramby resizing the overview's viewport tracker rectangle.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
get allowZoom() {}
set allowZoom(value) {}
```

Property Value

Boolean

true to let users zoom by resizing the tracker rectangle, or **false** otherwise.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview.autoScrollAmount Property

Gets or sets the amount by which to auto-scroll the view when the mouse leaves the view boundaries while drawing.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
get autoScrollAmount() {}
set autoScrollAmount(value) {}
```

Property Value

Number

A number specifying the amount by which to auto-scroll.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview.backColor Property

Gets or sets the background color of the overview control.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
get backColor() {}
set backColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview.diagramView Property

Gets or sets the DiagramView shown by this Overview.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
get diagramView() {}
set diagramView(value) {}
```

Property Value

DiagramView

The DiagramView shown by the overview.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview.dimColor Property

Gets or sets a color used to blend with the part of the diagram that is not visible in target DiagramView.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
get dimColor() {}
set dimColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview.scaleFactor Property

Gets or sets a fixed scale factor to apply when FitAll is disabled.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
get scaleFactor() {}
set scaleFactor(value) {}
```

Property Value

Number

A number specifying the scale as percentage.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview.scaleMode Property

Gets or sets a value indicating whether to fit the whole diagram inside the overview control.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
get scaleMode() {}
set scaleMode(value) {}
```

Property Value

ScaleMode

A member of the ScaleMode enumeration. The default value is **FitAll**.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview.scrollX Property

Gets the overview's horizontal scroll position.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
get scrollX() {}
set scrollX(value) {}
```

Property Value

Number

A number specifying the horizontal scroll position in current unit of measure.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview.scrollY Property

Gets the overview's vertical scroll position.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
get scrollY() {}
set scrollY(value) {}
```

Property Value

Number

A number specifying the vertical scroll position in current unit of measure.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview.viewport Property

Returns the visible portion of the diagram area.

Namespace: MindFusion.Diagramming
Assembly: Overview.js

Syntax

JavaScript Copy Code

```
get viewport() {}
set viewport(value) {}
```

Property Value

Rect

A Rect instance containing the viewport coordinates.

See Also

Overview Members
Overview Class
MindFusion.Diagramming Namespace

Overview Events

PageContainerCommand Class

Implements page navigation of PagedContainerNode.

Syntax

JavaScript Copy Code

```
// class
PageContainerCommand.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Command
MindFusion.Diagramming.PageContainerCommand

See Also

PageContainerCommand Members
MindFusion.Diagramming Namespace

PageContainerCommand Members

Implements page navigation of PagedContainerNode.

The following tables list the members exposed by the PageContainerCommand type.

Public Constructors

	Name	Description

	PageContainerCommand	Initializes a new instance of the PageContainerCommand class.

Public Methods

	Name	Description
	execute	Command.execute override.
	redo	Command.redo override.
	undo	Command.undo override.

See Also

PageContainerCommand Class
MindFusion.Diagramming Namespace

PageContainerCommand Constructor

Initializes a new instance of the PageContainerCommand class.

Namespace: MindFusion.Diagramming
Assembly: PageContainerCommand.js

Syntax

JavaScript Copy Code

```
function PageContainerCommand (container)
```

Parameters

container
PagedContainerNode. The PagedContainerNode modified by this command.

See Also

PageContainerCommand Class
MindFusion.Diagramming Namespace

PageContainerCommand Methods

PageContainerCommand.execute Method

Command.execute override.

Namespace: MindFusion.Diagramming
Assembly: PageContainerCommand.js

Syntax

JavaScript Copy Code

```
function execute ()
```

See Also

PageContainerCommand Members
PageContainerCommand Class
MindFusion.Diagramming Namespace

PageContainerCommand.redo Method

Command.redo override.

Namespace: MindFusion.Diagramming
Assembly: PageContainerCommand.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

PageContainerCommand Members
PageContainerCommand Class
MindFusion.Diagramming Namespace

PageContainerCommand.undo Method

Command.undo override.

Namespace: MindFusion.Diagramming
Assembly: PageContainerCommand.js

Syntax

JavaScript Copy Code

```
function undo ()
```

See Also

PageContainerCommand Members
PageContainerCommand Class
MindFusion.Diagramming Namespace

PagedContainerNode Class

A container that can arrange its children in pages.

Syntax

JavaScript Copy Code

```
// class
PagedContainerNode.prototype = {}
```

Remarks

PagedContainerNode is a ContainerNode subclass that organizes its child nodes into a collection of pages, represented by ContainerPage objects. This allows for the creation of more orgainzed diagrams where nodes can be grouped into logical views within a single container. Only the nodes belonging to the currently active page, specified by currentPage property, are rendered and participate in user interactions.

Navigation between pages is handled by scroll arrows in the caption bar, allowing users to cycle through the pages sequentially. For faster access, users can also double-click the container's caption to open an in-place combo box listing available page titles, enabling direct navigation to any page. The node's text automatically displays the title of newly activated **ContainerPage**.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.CaptionedNode

MindFusion.Diagramming.ContainerNode
MindFusion.Diagramming.PagedContainerNode

See Also

PagedContainerNode Members
MindFusion.Diagramming Namespace

PagedContainerNode Members

A container that can arrange its children in pages.

The following tables list the members exposed by the PagedContainerNode type.

Public Constructors

	Name	Description
	PagedContainerNode	Initializes a new instance of the PagedContainerNode class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowAddChildren	Gets or sets a value indicating whether child nodes can be added interactively to this container. (Inherited from ContainerNode.)
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AllowRemoveChildren	Gets or sets a value indicating whether child nodes can be removed interactively from this container. (Inherited from ContainerNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	AutoDeleteChildren	Gets or sets whether deleting this container should also remove its child items from the diagram. (Inherited from ContainerNode.)
	AutoGrow	Gets or sets a value indicating whether the container should grow in size when nodes are added to it. (Inherited from ContainerNode.)
	AutoShrink	Gets or sets a value indicating whether the container should shrink in size when nodes are removed from it. (Inherited from ContainerNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Caption	Gets or sets the text displayed as caption of the table. (Inherited from ContainerNode.)
	CaptionBackBrush	Gets or sets the brush used to paint the caption bar of this container. (Inherited from ContainerNode.)
	CaptionBrush	Gets or sets the brush used to paint the container caption. (Inherited from ContainerNode.)

	CaptionFormat	Gets or sets the caption alignment and format. (Inherited from ContainerNode.)
	CaptionHeight	Gets or sets the height of the container's caption bar. (Inherited from ContainerNode.)
	Children	Gets the child nodes of this container. (Inherited from ContainerNode.)
	ClipChildren	Gets or sets a value indicating whether child items should be clipped by container boundaries. (Inherited from ContainerNode.)
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent **ContainerNode** containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	currentPage	Gets or sets the current page.
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveCaptionBackBrush	Gets the effective caption back brush of the container. (Inherited from ContainerNode.)
	EffectiveCaptionBrush	Gets the effective caption brush of the container. (Inherited from ContainerNode.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled. (Inherited from ContainerNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Foldable	Gets or sets a value indicating whether the user is allowed to fold this container. (Inherited from ContainerNode.)
	Folded	Gets or sets a value indicating whether this container is folded. (Inherited from ContainerNode.)
	FoldedSize	Gets or sets the size of the container when folded. (Inherited from ContainerNode.)
	FoldIconSize	Gets or sets the size of the fold/unfold icon. (Inherited from ContainerNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HighlightPen	Gets or sets the pen used to draw the container frame lines when an item is dragged over the container. (Inherited from ContainerNode.)

	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	Margin	Gets or sets how much space should be left between contained nodes and the container borders. (Inherited from ContainerNode.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	MinimumSize	Gets or sets the minimum size of the container. (Inherited from ContainerNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	pages	Gets the pages of this container.
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Scrollable	Gets or sets a value indicating whether users are allowed to scroll the contents of this container. (Inherited from ContainerNode.)
	ScrollX	Gets or sets the container's horizontal scroll position. (Inherited from ContainerNode.)
	ScrollY	Gets or sets the container's vertical scroll position. (Inherited from ContainerNode.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)

	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Shape	Gets or sets the shape of the container's outline. (Inherited from ContainerNode.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	UnfoldedSize	Returns the original size of a folded container. (Inherited from ContainerNode.)
	Visible	Gets or sets a value indicating whether the container is visible. (Inherited from ContainerNode.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	Add	Adds a node to this container. (Inherited from ContainerNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	addPage	Adds a page to the container.

	Animate	Animates the item. (Inherited from DiagramItem.)
	Arrange	Arranges the container's children with the given layout. (Inherited from ContainerNode.)
	ArrangeAnimated	Arranges the container's children using the given layout, and animates items moving to their new positions. (Inherited from ContainerNode.)
	ArrangeAsync	Arranges the container's children with the given layout asynchronously. (Inherited from ContainerNode.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	clone	DiagramNode.clone override.
	Clone	DiagramItem.Clone override. Creates a copy of this item. (Inherited from ContainerNode.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	ContainsRecursively	Determines whether this container contains the specified node, either directly or recursively through other containers. (Inherited from ContainerNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramNode.Draw override. Draws the container on the specified IGraphics surface. (Inherited from ContainerNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	drawChildren	ContainerNode.drawChildren override.
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	DiagramNode.DrawLocal override. Draws the container on the specified IGraphics surface. (Inherited from ContainerNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified **IGraphics** surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	fromJson	DiagramNode.fromJson override.
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)

	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
S	GetContainer	Retrieves the container node (if any) the specified node belongs to. (Inherited from ContainerNode.)
	getEditRect	DiagramNode.getEditRect override.
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectiveCaptionBackBrush	Returns a Brush object representing the currently effective caption background of the container. (Inherited from ContainerNode.)
	GetEffectiveCaptionBrush	Returns a Brush object representing the currently effective caption brush of the container. (Inherited from ContainerNode.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramNode.GetRepaintRect override. (Inherited from ContainerNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	DiagramItem.HitTest override. (Inherited from ContainerNode.)
	HitTestHandle	DiagramNode.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from ContainerNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	insertPage	Inserts a page at the specified index.
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	loadFromXml	DiagramNode.loadFromXml override.
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	movePage	Moves a page to a new position in the page list.
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)

	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from ContainerNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from ContainerNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from ContainerNode.)
	OnLoad	DiagramItem.OnLoad override. (Inherited from ContainerNode.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	Remove	Removes a node from this container. (Inherited from ContainerNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	removePage	Removes a page from the container.
	Resize	Resizes the node. (Inherited from DiagramNode.)
	ResizeToFitChildren	Resizes the container making it big enough to contain its child nodes. (Inherited from ContainerNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveToXml	DiagramNode.saveToXml override.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	toJson	DiagramNode.toJson override.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	UpdateBounds	Overloaded. (Inherited from ContainerNode.)
	VisitNodes	Visits the diagram nodes in reverse Z order. (Inherited from ContainerNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from ContainerNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object. (Inherited from ContainerNode.)
	LoadFromXml	Loads the item content from an XML element. (Inherited from ContainerNode.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from ContainerNode.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramNode.OnRemove override. (Inherited from ContainerNode.)

	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramNode.RestoreProperties override. (Inherited from ContainerNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramNode.SaveProperties override. (Inherited from ContainerNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from ContainerNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from ContainerNode.)
	ScaleElements	DiagramNode.ScaleElements override. (Inherited from ContainerNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)

	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode Constructor

Initializes a new instance of the PagedContainerNode class.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript Copy Code

```
function PagedContainerNode ([parent, [pageTitles]])
```

Parameters

parent
Optional.
Diagram. The Diagram from which to obtain default values for the node properties.

pageTitles
Optional.
An array of title strings for the initial pages.

See Also

PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode Methods

PagedContainerNode.addPage Method

Adds a page to the container.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript Copy Code

```
function addPage (page)
```

Parameters

page
ContainerPage. The page to add.

Remarks

Container's current pages are accessible via pages property.

See Also

PagedContainerNode.clone Method

DiagramNode.clone override.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript Copy Code

```
function clone ()
```

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.drawChildren Method

ContainerNode.drawChildren override.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript Copy Code

```
function drawChildren ()
```

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.fromJson Method

DiagramNode.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript Copy Code

```
function fromJson ()
```

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.getEditRect Method

DiagramNode.getEditRect override.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript — Copy Code

```
function getEditRect ()
```

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.insertPage Method

Inserts a page at the specified index.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript — Copy Code

```
function insertPage (index, page)
```

Parameters

index
Number. The zero-based index at which the page should be inserted.

page
ContainerPage. The page to insert.

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.loadFromXml Method

DiagramNode.loadFromXml override.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript — Copy Code

```
function loadFromXml ()
```

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.movePage Method

Moves a page to a new position in the page list.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript Copy Code

```
function movePage (newIndex, page)
```

Parameters

newIndex
Number. The new index of the page.

page
ContainerPage. The page to move.

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.removePage Method

Removes a page from the container.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript Copy Code

```
function removePage (page)
```

Parameters

page
ContainerPage. The page to remove.

Remarks

Container's current pages are accessible via pages property.

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace
addPage Method

PagedContainerNode.saveToXml Method

DiagramNode.saveToXml override.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript Copy Code

```
function saveToXml ()
```

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.toJson Method

DiagramNode.toJson override.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode Properties

PagedContainerNode.currentPage Property

Gets or sets the current page.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript Copy Code

```
get currentPage() {}
set currentPage(value) {}
```

Property Value

ContainerPage

A ContainerPage reference.

Remarks

PagedContainerNode displays only child items associated with this page. Users can change the current page by clicking on arrow icons in the caption bar.

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.pages Property

Gets the pages of this container.

Namespace: MindFusion.Diagramming
Assembly: PagedContainerNode.js

Syntax

JavaScript — Copy Code

```
get pages() {}
```

Property Value

Array

An array of ContainerPage objects.

Remarks

Call addPage and removePage methods to modify this collection.

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PanBehavior Class

Pans the diagram when the users drags the mouse.

Syntax

JavaScript — Copy Code

```
// class
PanBehavior.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.BehaviorBase
MindFusion.Diagramming.PanBehavior

See Also

PanBehavior Members
MindFusion.Diagramming Namespace

PanBehavior Members

Pans the diagram when the users drags the mouse.

The following tables list the members exposed by the PanBehavior type.

Public Constructors

	Name	Description
	PanBehavior	Overloaded.

Public Properties

	Name	Description
	diagram	Gets the target diagram of this behavior object. (Inherited from BehaviorBase.)

Public Methods

	Name	Description

	createController	BehaviorBase.createController override.

See Also

PanBehavior Class
MindFusion.Diagramming Namespace

PanBehavior Constructor

Overload List

Name	Description
PanBehavior (Diagram)	Initializes a new instance of the PanBehavior class.
PanBehavior (DiagramView)	Initializes a new instance of the PanBehavior class.

See Also

PanBehavior Class
MindFusion.Diagramming Namespace

PanBehavior Constructor (Diagram)

Initializes a new instance of the PanBehavior class.

Namespace: MindFusion.Diagramming
Assembly: PanBehavior.js

Syntax

JavaScript Copy Code

```
function PanBehavior (parent)
```

Parameters

parent
The Diagram this behavior will be associated with.

See Also

PanBehavior Constructor Overload List
PanBehavior Class
MindFusion.Diagramming Namespace

PanBehavior Constructor (DiagramView)

Initializes a new instance of the PanBehavior class.

Namespace: MindFusion.Diagramming
Assembly: PanBehavior.js

Syntax

JavaScript Copy Code

```
function PanBehavior (parent)
```

Parameters

parent
DiagramView. The DiagramView this behavior will be associated with.

See Also

PanBehavior Constructor Overload List
PanBehavior Class
MindFusion.Diagramming Namespace

PanBehavior Methods

PanBehavior.createController Method

BehaviorBase.createController override.

Namespace: MindFusion.Diagramming
Assembly: PanBehavior.js

Syntax

JavaScript Copy Code

```
function createController (state)
```

Parameters

state
Object. An object representing current context and input event.

Return Value

SinglePointerController. An instance of SinglePointerController -derived class.

See Also

PanBehavior Members
PanBehavior Class
MindFusion.Diagramming Namespace

PanController Class

A controller used to pan the DiagramView.

Syntax

JavaScript Copy Code

```
// class
PanController.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.SinglePointerController
MindFusion.Diagramming.PanController

See Also

PanController Members
MindFusion.Diagramming Namespace

PanController Members

A controller used to pan the DiagramView.

The following tables list the members exposed by the PanController type.

Public Constructors

	Name	Description
	PanController	Overloaded.

Public Methods

	Name	Description
	cancel	Called when the user stops drawing or modifying a diagram element, rejecting the operation. (Inherited from SinglePointerController.)
	commit	Called when the user stops drawing or modifying a diagram element, accepting the operation. (Inherited from SinglePointerController.)
	drawInteraction	Called to draw current state of user interaction. (Inherited from SinglePointerController.)
	move	SinglePointerController.move override.
	start	SinglePointerController.start override.
	validate	Called to validate the operation at current pointer position. (Inherited from SinglePointerController.)

See Also

PanController Class
MindFusion.Diagramming Namespace

PanController Constructor

Overload List

Name	Description
PanController (Diagram)	Initializes a new instance of the PanController class.
PanController (DiagramView)	Initializes a new instance of the PanController class.

See Also

PanController Class
MindFusion.Diagramming Namespace

PanController Constructor (Diagram)

Initializes a new instance of the PanController class.

Namespace: MindFusion.Diagramming
Assembly: PanController.js

Syntax

JavaScript Copy Code

```
function PanController (diagram)
```

Parameters

diagram
The diagram that will get scrolled by this controller.

See Also

PanController Constructor Overload List
PanController Class
MindFusion.Diagramming Namespace

PanController Constructor (DiagramView)

Initializes a new instance of the PanController class.

Namespace: MindFusion.Diagramming
Assembly: PanController.js

Syntax

JavaScript Copy Code

```
function PanController (diagramView)
```

Parameters

diagramView
DiagramView. The diagramView that will get scrolled by this controller.

See Also

PanController Constructor Overload List
PanController Class
MindFusion.Diagramming Namespace

PanController Methods

PanController.move Method

SinglePointerController.move override.

Namespace: MindFusion.Diagramming
Assembly: PanController.js

Syntax

JavaScript Copy Code

```
function move (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

PanController Members
PanController Class
MindFusion.Diagramming Namespace

PanController.start Method

SinglePointerController.start override.

Namespace: MindFusion.Diagramming
Assembly: PanController.js

Syntax

JavaScript — Copy Code

```
function start (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

PanController Members
PanController Class
MindFusion.Diagramming Namespace

Panel Class

Represents layout containers in CompositeNode visual tree.

Syntax

JavaScript — Copy Code

```
// class
Panel.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.Container
  MindFusion.Diagramming.Panel
    MindFusion.Diagramming.GridPanel
    MindFusion.Diagramming.SimplePanel
    MindFusion.Diagramming.StackPanel

See Also

Panel Members
MindFusion.Diagramming Namespace
Composite Nodes

Panel Members

Represents layout containers in CompositeNode visual tree.

The following tables list the members exposed by the Panel type.

Public Constructors

	Name	Description
	Panel	Initializes a new instance of the Panel class.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)

	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

Public Methods

	Name	Description
	getComponents	Gets the child components of this panel.

See Also

Panel Class
MindFusion.Diagramming Namespace

Panel Constructor

Initializes a new instance of the Panel class.

Namespace: MindFusion.Diagramming
Assembly: Panel.js

Syntax

JavaScript Copy Code

```
function Panel ()
```

See Also

Panel Class
MindFusion.Diagramming Namespace

Panel Methods

Panel.getComponents Method

Gets the child components of this panel.

Namespace: MindFusion.Diagramming
Assembly: Panel.js

Syntax

JavaScript Copy Code

```
function getComponents ()
```

Return Value

Array. An array containing the child components of this panel.

See Also

Panel Members
Panel Class
MindFusion.Diagramming Namespace

Path Class

Represents a sequence of connected DiagramNode and DiagramLink elements.

Syntax

JavaScript Copy Code

```
// class
Path.prototype = {}
```

Remarks

Path instances are returned by methods of the PathFinder class.

Inheritance Hierarchy

MindFusion.Diagramming.Path

See Also

Path Members
MindFusion.Diagramming Namespace

Path Members

Represents a sequence of connected DiagramNode and DiagramLink elements.

The following tables list the members exposed by the Path type.

Public Constructors

	Name	Description
	Path	Initializes a new instance of the **Path** class.

Public Fields

	Name	Description
	items	An array of all DiagramItem elements participating in the path.
	links	An array of all **DiagramLink** elements participating in the path.
	nodes	An array of all **DiagramNode** elements participating in the path.

Public Methods

	Name	Description
	getLength	Gets the length of the **Path** expressed as the number of **links** participating in it.
	getWeight	Calculates the weight of this **Path** by accumulating the Weight of all participating diagram elements.

See Also

Path Class
MindFusion.Diagramming Namespace

Path Fields

Path.items Field

An array of all DiagramItem elements participating in the path.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
var items
```

Remarks

The collection contains consecutive DiagramNode and DiagramLink objects.

See Also

Path Members
Path Class
MindFusion.Diagramming Namespace

Path.links Field

An array of all DiagramLink elements participating in the path.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
var links
```

Remarks

Use nodes to get all DiagramNode elements in the Path.

See Also

Path Members
Path Class
MindFusion.Diagramming Namespace

Path.nodes Field

An array of all DiagramNode elements participating in the path.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
var nodes
```

Remarks

Use links to get all DiagramLink elements in the Path.

See Also

Path Constructor

Initializes a new instance of the Path class.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
function Path ()
```

See Also

Path Class
MindFusion.Diagramming Namespace

Path Methods

Path.getLength Method

Gets the length of the Path expressed as the number of links participating in it.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
function getLength ()
```

Return Value

Number. An integer value that represents the number of edges (links) in the path.

See Also

Path Members
Path Class
MindFusion.Diagramming Namespace

Path.getWeight Method

Calculates the weight of this Path by accumulating the Weight of all participating diagram elements.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
function getWeight ([incNodes, [incLinks]])
```

Parameters

incNodes
Optional.

A boolean value specifying whether the Weight of DiagramNode elements should be considered.

incLinks
Optional.
A boolean value specifying whether the Weight of DiagramLink elements should be considered.

Return Value

A number that represents the total weight of the elements in this Path according to the *incNodes* and *incLinks* parameters.

See Also

Path Members
Path Class
MindFusion.Diagramming Namespace

PathFinder Class

Provides methods that help you find paths and cycles in a graph.

Syntax

JavaScript Copy Code

```
// class
PathFinder.prototype = {}
```

Remarks

The direction of links is taken into account when looking for paths or cycles.

A path-finder object caches graph information between method calls in order to achieve better performance when path and cycle -finding methods are called on the same diagram more than once. If the diagram structure changes, you must create a new PathFinder instance in order to use the new structure of diagram's graph.

Inheritance Hierarchy

MindFusion.Diagramming.PathFinder

See Also

PathFinder Members
MindFusion.Diagramming Namespace

PathFinder Members

Provides methods that help you find paths and cycles in a graph.

The following tables list the members exposed by the PathFinder type.

Public Constructors

	Name	Description
	PathFinder	Initializes a new instance of the PathFinder class.

Public Methods

	Name	Description
	findAllCycles	Finds all cycles in the underlying diagram.
	findAllPaths	Finds and returns all paths that exist between two DiagramNode objects.
	findCycle	Detects whether the specified **DiagramNode** participates in a cycle.
	findLongestPath	Finds the longest path between the specified **DiagramNode** objects.

	findShortestPath	Finds the shortest path between two **DiagramNode** elements, optionally considering the Weight of nodes and / or links.
	findShortestRoute	Calculates a route that passes through all nodes of the diagram, using only existing links. This implementation of the Traveling Salesperson Problem uses the nearest-neighbor heuristic.
	generateShortestRoute	Calculates a route that passes through all nodes of the diagram. This implementation of the Traveling Salesperson Problem uses the nearest-neighbor heuristic.

See Also

PathFinder Class
MindFusion.Diagramming Namespace

PathFinder Constructor

Initializes a new instance of the PathFinder class.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
function PathFinder (diagram)
```

Parameters

diagram
An instance of the Diagram class in which to look for paths or cycles.

See Also

PathFinder Class
MindFusion.Diagramming Namespace

PathFinder Methods

PathFinder.findAllCycles Method

Finds all cycles in the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
function findAllCycles ()
```

Return Value

An array of Path objects; if no cycles are detected, the array is empty.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
findCycle Method

PathFinder.findAllPaths Method

Finds and returns all paths that exist between two DiagramNode objects.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript — Copy Code

```
function findAllPaths (from, to, [maxPaths])
```

Parameters

from
The first DiagramNode in the Path.

to
The last **DiagramNode** in the **Path**.

maxPaths
Optional.
Number. The maximum number of paths to find.

Return Value

An array of **Path** objects; if no paths are found, the array is empty.

Remarks

The returned Path instances include the *from* and *to* nodes.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace

PathFinder.findCycle Method

Detects whether the specified DiagramNode participates in a cycle.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript — Copy Code

```
function findCycle (participant)
```

Parameters

participant
A DiagramNode element which should participate in the found cycle.

Return Value

A Path instance that represents the found cycle; **null** if no cycle is detected.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
findAllCycles Method

PathFinder.findLongestPath Method

Finds the longest path between the specified DiagramNode objects.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript — Copy Code

```
function findLongestPath ([from, [to]])
```

Parameters

from
Optional.
The first DiagramNode element in the Path.

to
Optional.
The last **DiagramNode** element in the **Path**.

Return Value

A **Path** instance that represents the longest path found; **null** if there is no **Path** connecting *from* and *to*.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
findAllPaths Method

PathFinder.findShortestPath Method

Finds the shortest path between two DiagramNode elements, optionally considering the Weight of nodes and / or links.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript — Copy Code

```
function findShortestPath (from, to, [useNodeWeights, [useLinkWeights]])
```

Parameters

from
The first DiagramNode in the Path.

to
The last **DiagramNode** in the **Path**.

useNodeWeights
Optional.
If **true**, the **Path** with minimal total weight is found, summing the Weight of nodes in the **Path**.

useLinkWeights
Optional.
If **true**, the **Path** with minimal total weight is found, summing the Weight of links in the **Path**.

Return Value

A **Path** instance that represents the shortest **Path** found or **null** if there is no **Path** connecting *from* and *to*.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
findAllPaths Method

PathFinder.findShortestRoute Method

Calculates a route that passes through all nodes of the diagram, using only existing links. This implementation of the Traveling Salesperson Problem uses the nearest-neighbor heuristic.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
function findShortestRoute ()
```

Return Value

A Path object with its nodes, links and items fields populated, representing the approximate shortest route found by the heuristic.

Remarks

This method expects to be called on a connected graph and builds a path consisting of existing diagram links. The weight of the links is taken from their weight property.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
generateShortestRoute Method

PathFinder.generateShortestRoute Method

Calculates a route that passes through all nodes of the diagram. This implementation of the Traveling Salesperson Problem uses the nearest-neighbor heuristic.

Namespace: MindFusion.Diagramming
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
function generateShortestRoute ([weight])
```

Parameters

weight
Optional.
function. A function that takes two DiagramNode arguments and returns a number that represents the distance between them. If not provided, the geometric distance between the nodes' centers is used.

Return Value

A Path object with its nodes and items fields populated, representing approximate shortest route found by the heuristic.

Remarks

This method does not consider current links in the diagram. Instead, it uses geometrical distance between nodes or the result of *weight* function to suggest the sequence of nodes you should connect to generate approximate shortest path passing between them.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
findShortestRoute Method

PatternRouter Class

Routes links by applying patterns of segment and turn sequences and accepting them when resulting paths do not cross nodes.

Syntax

JavaScript Copy Code

```
// class
PatternRouter.prototype = {}
```

Remarks

You can add custom patterns to the patterns property, or replace the standard patterns altogether. Patterns are defined using RoutePattern and RouteStep objects.

Example

The following code defines a Z-like link route that will be accepted if it does not cross obstacle nodes, nor passes by them within minimumDistance.

JavaScript Copy Code

```
var rightZigzag = new RoutePattern();
rightZigzag.steps.push(
    new RouteStep(RouteStepKind.AdvanceRelative, 0.5));
rightZigzag.steps.push(
    new RouteStep(RouteStepKind.TurnRight));
rightZigzag.steps.push(
    new RouteStep(RouteStepKind.AdvanceRelative, 1));
rightZigzag.steps.push(
    new RouteStep(RouteStepKind.TurnLeft));
rightZigzag.steps.push(
    new RouteStep(RouteStepKind.AdvanceRelative, 0.5));
router.patterns.push(rightZigzag);

// this can be defined in shorter form using a pattern string
// var rightZigzag = new RoutePattern(
//     "ADR0.5 TRR ADR1 TRL ADR0.5");
```

Inheritance Hierarchy

MindFusion.Diagramming.Router
  MindFusion.Diagramming.PatternRouter

See Also

PatternRouter Members
MindFusion.Diagramming Namespace

PatternRouter Members

Routes links by applying patterns of segment and turn sequences and accepting them when resulting paths do not cross nodes.

The following tables list the members exposed by the PatternRouter type.

Public Constructors

	Name	Description
	PatternRouter	Initializes a new instance of the PatternRouter class.

Public Properties

	Name	Description

	avoidOverlaps	Gets or sets a value indicating whether to avoid overlapping links. (Inherited from Router.)
	minimumDistance	Gets or sets the minimum allowed distance from obstacles for a route to be accepted.
	patterns	Gets or sets a list of RoutePattern objects representing the routing patterns.
	preferredDistance	Gets or sets the preferred distance from link's bend points to the link's nodes.

Public Methods

	Name	Description
	route	Called when a single link must be routed. (Inherited from Router.)
	routeInternal	Overrides Router.routeInternal.
	routeLinks	Called when more than one link must be routed. (Inherited from Router.)

See Also

PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter Constructor

Initializes a new instance of the PatternRouter class.

Namespace: MindFusion.Diagramming
Assembly: PatternRouter.js

Syntax

JavaScript Copy Code

```
function PatternRouter ([diagram])
```

Parameters

diagram
Optional.
Diagram. The diagram this router is associated with.

See Also

PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter Methods

PatternRouter.routeInternal Method

Overrides Router.routeInternal.

Namespace: MindFusion.Diagramming
Assembly: PatternRouter.js

Syntax

JavaScript Copy Code

```
function routeInternal (link)
```

Parameters

link
DiagramLink. The link to route.

Return Value

Array. An array of points representing the link's new shape, or null if routing fails.

See Also

PatternRouter Members
PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter Properties

PatternRouter.minimumDistance Property

Gets or sets the minimum allowed distance from obstacles for a route to be accepted.

Namespace: MindFusion.Diagramming
Assembly: PatternRouter.js

Syntax

JavaScript Copy Code

```
get minimumDistance() {}
set minimumDistance(value) {}
```

Property Value

Number

A number specifying minimum distance. The default value is 4.

See Also

PatternRouter Members
PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter.patterns Property

Gets or sets a list of RoutePattern objects representing the routing patterns.

Namespace: MindFusion.Diagramming
Assembly: PatternRouter.js

Syntax

JavaScript Copy Code

```
get patterns() {}
set patterns(value) {}
```

Property Value

Array

An array of RoutePattern objects defining link route templates.

See Also

PatternRouter Members
PatternRouter Class

PatternRouter.preferredDistance Property

Gets or sets the preferred distance from link's bend points to the link's nodes.

Namespace: MindFusion.Diagramming
Assembly: PatternRouter.js

Syntax

JavaScript Copy Code

```
get preferredDistance() {}
set preferredDistance(value) {}
```

Property Value

Number

A number specifying specifying preferred distance. The default value is 10.

Remarks

The value of this property can be used in route templates using an AdvancePreferred RouteStep.

See Also

PatternRouter Members
PatternRouter Class
MindFusion.Diagramming Namespace

RelativeToLink Enumeration

Specifies how the link labels are positioned.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
RelativeToLink = {}
```

Members

	Member name	Description
	Segment	The link labels are positioned relative to a specified segment.
	ControlPoint	The link labels are positioned relative to a specified control point.
	LinkLength	The link labels are positioned relative to the link.

Remarks

Members of this enumeration can be passed as arguments to LinkLabel positioning methods.

See Also

MindFusion.Diagramming Namespace

RelativeToNode Enumeration

Specifies how the node labels are positioned.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
RelativeToNode = {}
```

Members

	Member name	Description
	Edge	The node labels are positioned relative to a specified edge.
	Corner	The node labels are positioned relative to a specified corner point.
	Center	The node labels are positioned relative to the node center.

See Also

MindFusion.Diagramming Namespace

RemoveItemCommand Class

Implements deletion of diagram items.

Syntax

JavaScript — Copy Code

```
// class
RemoveItemCommand.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Command
  MindFusion.Diagramming.RemoveItemCommand

See Also

RemoveItemCommand Members
MindFusion.Diagramming Namespace

RemoveItemCommand Members

Implements deletion of diagram items.

The following tables list the members exposed by the RemoveItemCommand type.

Public Constructors

	Name	Description
	RemoveItemCommand	Initializes a new instance of the **RemoveItemCommand** class.

Public Methods

	Name	Description
	execute	Command.execute override.
	redo	Command.redo override.
	undo	Command.undo override.

See Also

RemoveItemCommand Class
MindFusion.Diagramming Namespace

RemoveItemCommand Constructor

Initializes a new instance of the RemoveItemCommand class.

Namespace: MindFusion.Diagramming
Assembly: RemoveItemCommand.js

Syntax

JavaScript Copy Code

```
function RemoveItemCommand (diagram, item)
```

Parameters

diagram
The Diagram modified by this command.

item
A DiagramItem that should be removed from the **Diagram**.

See Also

RemoveItemCommand Class
MindFusion.Diagramming Namespace

RemoveItemCommand Methods

RemoveItemCommand.execute Method

Command.execute override.

Namespace: MindFusion.Diagramming
Assembly: RemoveItemCommand.js

Syntax

JavaScript Copy Code

```
function execute ()
```

See Also

RemoveItemCommand Members
RemoveItemCommand Class
MindFusion.Diagramming Namespace

RemoveItemCommand.redo Method

Command.redo override.

Namespace: MindFusion.Diagramming
Assembly: RemoveItemCommand.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

RemoveItemCommand Members
RemoveItemCommand Class
MindFusion.Diagramming Namespace

RemoveItemCommand.undo Method

Command.undo override.

Namespace: MindFusion.Diagramming
Assembly: RemoveItemCommand.js

Syntax

JavaScript Copy Code

```
function undo ()
```

See Also

RemoveItemCommand Members
RemoveItemCommand Class
MindFusion.Diagramming Namespace

ResizeTableColumnCommand Class

Implements resizing of table's columns.

Syntax

JavaScript Copy Code

```
// class
ResizeTableColumnCommand.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Command
  MindFusion.Diagramming.ResizeTableColumnCommand

See Also

ResizeTableColumnCommand Members
MindFusion.Diagramming Namespace

ResizeTableColumnCommand Members

Implements resizing of table's columns.

The following tables list the members exposed by the ResizeTableColumnCommand type.

Public Constructors

	Name	Description
	ResizeTableColumnCommand	Initializes a new instance of the ResizeTableColumnCommand class.

Public Methods

	Name	Description
	execute	Command.execute override.
	redo	Command.redo override.
	undo	Command.undo override.

See Also

ResizeTableColumnCommand Class
MindFusion.Diagramming Namespace

ResizeTableColumnCommand Constructor

Initializes a new instance of the ResizeTableColumnCommand class.

Namespace: MindFusion.Diagramming
Assembly: ResizeTableColumnCommand.js

Syntax

JavaScript Copy Code

```
function ResizeTableColumnCommand (diagram, table, column)
```

Parameters

diagram
Diagram. The Diagram modified by this command.

table
TableNode. The TableNode instance that will be modified.

column
Object. The column object that will be resized.

See Also

ResizeTableColumnCommand Class
MindFusion.Diagramming Namespace

ResizeTableColumnCommand Methods

ResizeTableColumnCommand.execute Method

Command.execute override.

Namespace: MindFusion.Diagramming
Assembly: ResizeTableColumnCommand.js

Syntax

JavaScript Copy Code

```
function execute ()
```

See Also

ResizeTableColumnCommand Members
ResizeTableColumnCommand Class
MindFusion.Diagramming Namespace

ResizeTableColumnCommand.redo Method

Command.redo override.

Namespace: MindFusion.Diagramming
Assembly: ResizeTableColumnCommand.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

ResizeTableColumnCommand Members
ResizeTableColumnCommand Class
MindFusion.Diagramming Namespace

ResizeTableColumnCommand.undo Method

Command.undo override.

Namespace: MindFusion.Diagramming
Assembly: ResizeTableColumnCommand.js

Syntax

JavaScript Copy Code

```
function undo ()
```

See Also

ResizeTableColumnCommand Members
ResizeTableColumnCommand Class
MindFusion.Diagramming Namespace

ResizeTableRowCommand Class

Implements resizing of table's rows.

Syntax

JavaScript Copy Code

```
// class
ResizeTableRowCommand.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Command
  MindFusion.Diagramming.ResizeTableRowCommand

See Also

ResizeTableRowCommand Members
MindFusion.Diagramming Namespace

ResizeTableRowCommand Members

Implements resizing of table's rows.

The following tables list the members exposed by the ResizeTableRowCommand type.

Public Constructors

	Name	Description
	ResizeTableRowCommand	Initializes a new instance of the ResizeTableRowCommand class.

Public Methods

	Name	Description
	execute	Command.execute override.
	redo	Command.redo override.
	undo	Command.undo override.

See Also

ResizeTableRowCommand Class
MindFusion.Diagramming Namespace

ResizeTableRowCommand Constructor

Initializes a new instance of the ResizeTableRowCommand class.

Namespace: MindFusion.Diagramming
Assembly: ResizeTableRowCommand.js

Syntax

JavaScript Copy Code

```
function ResizeTableRowCommand (diagram, table, row)
```

Parameters

diagram
Diagram. The Diagram modified by this command.

table
TableNode. The TableNode instance that will be modified.

row
Object. The row object that will be resized.

See Also

ResizeTableRowCommand Class
MindFusion.Diagramming Namespace

ResizeTableRowCommand Methods

ResizeTableRowCommand.execute Method

Command.execute override.

Namespace: MindFusion.Diagramming
Assembly: ResizeTableRowCommand.js

Syntax

JavaScript Copy Code

```
function execute ()
```

See Also

ResizeTableRowCommand Members
ResizeTableRowCommand Class
MindFusion.Diagramming Namespace

ResizeTableRowCommand.redo Method

Command.redo override.

Namespace: MindFusion.Diagramming
Assembly: ResizeTableRowCommand.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

ResizeTableRowCommand Members
ResizeTableRowCommand Class
MindFusion.Diagramming Namespace

ResizeTableRowCommand.undo Method

Command.undo override.

Namespace: MindFusion.Diagramming
Assembly: ResizeTableRowCommand.js

Syntax

JavaScript Copy Code

```
function undo ()
```

See Also

ResizeTableRowCommand Members
ResizeTableRowCommand Class
MindFusion.Diagramming Namespace

RoutePattern Class

Represents a template from which routes are created.

Syntax

JavaScript Copy Code

```
// class
RoutePattern.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.RoutePattern

See Also

RoutePattern Members
MindFusion.Diagramming Namespace

RoutePattern Members

Represents a template from which routes are created.

The following tables list the members exposed by the RoutePattern type.

Public Properties

	Name	Description
	steps	A list of RouteStep objects that generate a route.

See Also

RoutePattern Class
MindFusion.Diagramming Namespace

RoutePattern Properties

RoutePattern.steps Property

A list of RouteStep objects that generate a route.

Namespace: MindFusion.Diagramming
Assembly: PatternRouter.js

Syntax

JavaScript Copy Code

```
get steps() {}
set steps(value) {}
```

Property Value

Array

An array of RouteStep objects.

See Also

RoutePattern Members
RoutePattern Class
MindFusion.Diagramming Namespace

Router Class

Implements a simple link routing algorithm that pulls link segments out of nodes they would overlap.

Syntax

JavaScript Copy Code

```
// class
Router.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Router
MindFusion.Diagramming.CompositeRouter
MindFusion.Diagramming.GridRouter
MindFusion.Diagramming.PatternRouter

See Also

Router Members
MindFusion.Diagramming Namespace
autoRoute Property
routeAllLinks Method
obstacle Property

Router Members

Implements a simple link routing algorithm that pulls link segments out of nodes they would overlap.

The following tables list the members exposed by the Router type.

Public Constructors

	Name	Description
	Router	Initializes a new instance of the Router class.

Public Properties

	Name	Description
	avoidOverlaps	Gets or sets a value indicating whether to avoid overlapping links.

Public Methods

	Name	Description
	route	Called when a single link must be routed.
	routeInternal	Calculates actual route and returns it as an array of points.
	routeLinks	Called when more than one link must be routed.

See Also

Router Class
MindFusion.Diagramming Namespace

Router Constructor

Initializes a new instance of the Router class.

Namespace: MindFusion.Diagramming
Assembly: Router.js

Syntax

JavaScript Copy Code

```
function Router ([diagram])
```

Parameters

diagram
Optional.
Diagram. The diagram this router is associated with.

See Also

Router Class
MindFusion.Diagramming Namespace

Router Methods

Router.route Method

Called when a single link must be routed.

Namespace: MindFusion.Diagramming
Assembly: Router.js

Syntax

JavaScript Copy Code

```
function route (link)
```

Parameters

link
DiagramLink. The link to route.

See Also

Router Members
Router Class
MindFusion.Diagramming Namespace

Router.routeInternal Method

Calculates actual route and returns it as an array of points.

Namespace: MindFusion.Diagramming
Assembly: Router.js

Syntax

JavaScript Copy Code

```
function routeInternal (link)
```

Parameters

link
DiagramLink. The link to route.

Return Value

Array.<Point>. An array of points representing the link's new shape, or null if routing fails.

See Also

Router Members
Router Class
MindFusion.Diagramming Namespace

Router.routeLinks Method

Called when more than one link must be routed.

Namespace: MindFusion.Diagramming
Assembly: Router.js

Syntax

JavaScript — Copy Code

```
function routeLinks (links)
```

Parameters

links
Array.<DiagramLink>. A list of links to route.

See Also

Router Members
Router Class
MindFusion.Diagramming Namespace

Router Properties

Router.avoidOverlaps Property

Gets or sets a value indicating whether to avoid overlapping links.

Namespace: MindFusion.Diagramming
Assembly: Router.js

Syntax

JavaScript — Copy Code

```
get avoidOverlaps() {}
set avoidOverlaps(value) {}
```

Property Value

boolean

true to avoid link-link overlaps, otherwise false.

See Also

Router Members
Router Class
MindFusion.Diagramming Namespace

RouteStep Class

Represents segments or turns in routing patterns.

Syntax

JavaScript — Copy Code

```
// class
RouteStep.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.RouteStep

See Also

RouteStep Members
MindFusion.Diagramming Namespace

RouteStep Members

Represents segments or turns in routing patterns.

The following tables list the members exposed by the RouteStep type.

Public Constructors

	Name	Description
	RouteStep	Initializes a new instance of the **RouteStep** class.

See Also

RouteStep Class
MindFusion.Diagramming Namespace

RouteStep Constructor

Initializes a new instance of the RouteStep class.

Namespace: MindFusion.Diagramming
Assembly: PatternRouter.js

Syntax

JavaScript Copy Code

```
function RouteStep (kind, [amount])
```

Parameters

kind
RouteStepKind. A member of the **RouteStepKind** enumeration.

amount
Optional.
Number. The distance to add when step kind is Advance*.

See Also

RouteStep Class
MindFusion.Diagramming Namespace

RouteStepKind Enumeration

Represents parts of a route template.

Namespace: MindFusion.Diagramming
Assembly: PatternRouter.js

Syntax

JavaScript Copy Code

```
// enum
RouteStepKind = {}
```

Members

	Member name	Description
	AdvanceRelative	Advance by the specified relative distance from origin to destination node.Equivalent to "ADR" command in pattern strings.
	AdvanceAbsolute	Advance by the specified absolute distance.Corresponds to "ADA" command in pattern strings.
	AdvancePreferred	Advance by PreferredDistance.Corresponds to "ADP" command in pattern strings.
	AdvanceDest	Advance by the specified relative remaining distance to destination node.Corresponds to "ADD" command in pattern strings.
	AdvanceRemaining	Advance by the specified relative remaining distance to link's end point.Corresponds to "ADM" command in pattern strings.
	AlignToOrigin	Advance to align with nearest side of origin node.Corresponds to "ALO" command in pattern strings.
	AlignToDestination	Advance to align with nearest side of destination node.Corresponds to "ALD" command in pattern strings.
	AlignToStart	Advance to align with link's start point.Corresponds to "ALS" command in pattern strings.
	AlignToEnd	Advance to align with link's end point.Corresponds to "ALE" command in pattern strings.
	AlignToFarBorder	Advance to align with furthest border of both nodes.Corresponds to "AFB" command in pattern strings.
	TurnLeft	Turn 90 degrees to the left.Corresponds to "TRL" command in pattern strings.
	TurnRight	Turn 90 degrees to the right.Corresponds to "TRR" command in pattern strings.

See Also

MindFusion.Diagramming Namespace

Ruler Class

The **Ruler** control provides horizontal and vertical scales that help users measure and align diagram items.

Syntax

JavaScript Copy Code

```
// class
Ruler.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.Control
  MindFusion.Diagramming.Ruler

See Also

Ruler Members
MindFusion.Diagramming Namespace

Ruler Members

The Ruler control provides horizontal and vertical scales that help users measure and align diagram items.

The following tables list the members exposed by the **Ruler** type.

Public Constructors

	Name	Description
	Ruler	Initializes a new instance of the Ruler class.

Public Properties

	Name	Description
	backColor	Gets or sets the background color of the control.
	diagramView	Gets or sets the DiagramView shown by this Ruler.
	element	Returns a reference to the control's DOM element. (Inherited from Control.)
	enabled	Gets or sets whether mouse events are enabled. (Inherited from Control.)
	enableGuides	Gets or sets a value indicating whether the alignment guides are enabled.
	foreColor	Gets or sets the foreground color of the control.
	guideColor	Gets or sets the color of the alignment guide lines.
	horizontalScaleVisible	Gets or sets a value indicating whether the horizontal ruler scale is visible.
	licenseLocation	Gets or sets the location of the license key file. (Inherited from Control.)
	negatedX	Gets or sets whether the horizontal scale should show negated values, simulating that the coordinate system's X axis grows to the left.
	negatedY	Gets or sets whether the vertical scale should show negated values, simulating that the coordinate system's Y axis grows to the top.
	pointerColor	Gets or sets the color of the mouse pointer's projections.
	projectionColor	Gets or sets the color used to paint the projection of the active node.
	projectRotatedBounds	Gets or sets whether projections of nodes on ruler's scales reflect their rotated boundaries.
	textColor	Gets or sets the color used to paint text in the ruler.
	unit	Gets the measure units displayed in the ruler scales.
	verticalScaleVisible	Gets or sets a value indicating whether the vertical ruler scale is visible.

Public Methods

	Name	Description
	addEventListener	Registers a single event listener on the instance. (Inherited from Disposable.)
S	addHandlers	Adds an event listener to a DOM element. (Inherited from Control.)
S	clearHandlers	Removes all event listeners from a DOM element. (Inherited from Control.)
S	create	Creates and initializes a new Ruler from the specified element. This method is static and can be called without creating an instance of the class.

	find	Returns the specified Ruler object. This member is static and can be invoked without creating an instance of the class.
	fromJson	Constructs a JavaScript object from a JSON string. (Inherited from Control.)
	raiseEvent	Raises an event. (Inherited from Disposable.)
	registerForSubmit	Registers an onsubmit handler for the Ruler's parent html form to flush postback data.
	removeEventListener	Removes a single event listener attached to the instance. (Inherited from Disposable.)
	toJson	Converts a JavaScript object to a JSON string. (Inherited from Control.)

Public Events

	Name	Description
	controlLoaded	Raised when the control is loaded. (Inherited from Control.)
	controlUnloaded	Raised when the control is unloaded. (Inherited from Control.)
	controlUnloading	Raised when the control is unloading. (Inherited from Control.)

See Also

Ruler Class
MindFusion.Diagramming Namespace

Ruler Constructor

Initializes a new instance of the Ruler class.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
function Ruler (element)
```

Parameters

element
HTMLDivElement. The Canvas DOM Element this Ruler is associated with.

See Also

Ruler Class
MindFusion.Diagramming Namespace

Ruler Methods

Ruler.create Method

Creates and initializes a new Ruler from the specified element. This method is static and can be called without creating an instance of the class.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
function create (element)
```

Parameters

element
HTMLDivElement. The DOM element that the ruler should be attached to.

Return Value

A Ruler object that represents the newly created ruler.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.find Method

Returns the specified Ruler object. This member is static and can be invoked without creating an instance of the class.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
function find (id)
```

Parameters

id
String. The id of the Ruler's element.

Return Value

A Ruler object with the specified id, if found, or **null** otherwise.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.registerForSubmit Method

Registers an onsubmit handler for the Ruler's parent html form to flush postback data.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
function registerForSubmit (id)
```

Parameters

id
String. The id of the hidden field to flush the data to.

See Also

Ruler Properties

Ruler.backColor Property

Gets or sets the background color of the control.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get backColor() {}
set backColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.diagramView Property

Gets or sets the DiagramView shown by this Ruler.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get diagramView() {}
set diagramView(value) {}
```

Property Value

DiagramView

The DiagramView attached to this ruler.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.enableGuides Property

Gets or sets a value indicating whether the alignment guides are enabled.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get enableGuides() {}
set enableGuides(value) {}
```

Property Value

Boolean

true to allow aligning diagram items using ruler guides, or **false** otherwise. The default is **true**.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.foreColor Property

Gets or sets the foreground color of the control.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get foreColor() {}
set foreColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.guideColor Property

Gets or sets the color of the alignment guide lines.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get guideColor() {}
set guideColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Ruler.horizontalScaleVisible Property

Gets or sets a value indicating whether the horizontal ruler scale is visible.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get horizontalScaleVisible() {}
set horizontalScaleVisible(value) {}
```

Property Value

Boolean

true to display the horizontal scale, or **false** otherwise. The default is **true**.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.negatedX Property

Gets or sets whether the horizontal scale should show negated values, simulating that the coordinate system's X axis grows to the left.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get negatedX() {}
set negatedX(value) {}
```

Property Value

Boolean

true if the horizontal scale should show negated values, or **false** otherwise. The default value is **false**.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.negatedY Property

Gets or sets whether the vertical scale should show negated values, simulating that the coordinate system's Y axis grows to the top.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get negatedY() {}
set negatedY(value) {}
```

Property Value

Boolean

true if the vertical scale should show negated values, or **false** otherwise. The default value is **false**.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.pointerColor Property

Gets or sets the color of the mouse pointer's projections.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get pointerColor() {}
set pointerColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.projectionColor Property

Gets or sets the color used to paint the projection of the active node.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get projectionColor() {}
set projectionColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.projectRotatedBounds Property

Gets or sets whether projections of nodes on ruler's scales reflect their rotated boundaries.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get projectRotatedBounds() {}
set projectRotatedBounds(value) {}
```

Property Value

Boolean

true if node's projection should match the boundaries of rotated node shape, or **false** to ignore rotation and project only the coordinates of node's Bounds. The default value is **true**.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.textColor Property

Gets or sets the color used to paint text in the ruler.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.unit Property

Gets the measure units displayed in the ruler scales.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get unit() {}
set unit(value) {}
```

Property Value

GraphicsUnit

A member of the GraphicsUnit enumeration.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler.verticalScaleVisible Property

Gets or sets a value indicating whether the vertical ruler scale is visible.

Namespace: MindFusion.Diagramming
Assembly: Ruler.js

Syntax

JavaScript Copy Code

```
get verticalScaleVisible() {}
set verticalScaleVisible(value) {}
```

Property Value

Boolean

true to display the vertical scale, or **false** otherwise. The default is **true**.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming Namespace

Ruler Events

SaveToStringFormat Enumeration

Specifies in what format to save the diagram when generating a string using the saveToString method.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
SaveToStringFormat = {}
```

Members

	Member name	Description
	Json	Indicates JSON format.
	Xml	Indicates XML format.

See Also

MindFusion.Diagramming Namespace

ScaleMode Enumeration

Defines Overview scaling modes.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
ScaleMode = {}
```

Members

	Member name	Description
	FitAll	Scale the Overview dynamically to fit the entire diagram.
	CombinedScales	Combine diagram's ZoomFactor with overview's ScaleFactor.
	FixedScale	Use fixed scale specified via ScaleFactor, ignoring diagram's zoom level.

Remarks

Members of this enumeration can be assigned to the scaleMode property of **Overview**.

See Also

MindFusion.Diagramming Namespace

ScrollTableCommand Class

Handles undo / redo of table scrolling.

Syntax

JavaScript — Copy Code

```
// class
ScrollTableCommand.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Command
  MindFusion.Diagramming.ScrollTableCommand

See Also

ScrollTableCommand Members
MindFusion.Diagramming Namespace

ScrollTableCommand Members

Handles undo / redo of table scrolling.

The following tables list the members exposed by the ScrollTableCommand type.

Public Constructors

	Name	Description
	ScrollTableCommand	Initializes a new instance of the **ScrollTableCommand** class.

Public Methods

	Name	Description
	execute	Command.execute override.
	redo	Command.redo override.
	undo	Command.undo override.

See Also

ScrollTableCommand Class
MindFusion.Diagramming Namespace

ScrollTableCommand Constructor

Initializes a new instance of the ScrollTableCommand class.

Namespace: MindFusion.Diagramming
Assembly: ScrollTableCommand.js

Syntax

JavaScript Copy Code

```
function ScrollTableCommand (diagram, table)
```

Parameters

diagram
The Diagram modified by this command.

table
The TableNode instance that will be modified.

See Also

ScrollTableCommand Class
MindFusion.Diagramming Namespace

ScrollTableCommand Methods

ScrollTableCommand.execute Method

Command.execute override.

Namespace: MindFusion.Diagramming
Assembly: ScrollTableCommand.js

Syntax

JavaScript Copy Code

```
function execute ()
```

See Also

ScrollTableCommand Members
ScrollTableCommand Class
MindFusion.Diagramming Namespace

ScrollTableCommand.redo Method

Command.redo override.

Namespace: MindFusion.Diagramming
Assembly: ScrollTableCommand.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

ScrollTableCommand Members
ScrollTableCommand Class
MindFusion.Diagramming Namespace

ScrollTableCommand.undo Method

Command.undo override.

Namespace: MindFusion.Diagramming
Assembly: ScrollTableCommand.js

Syntax

JavaScript Copy Code

```
function undo ()
```

See Also

ScrollTableCommand Members
ScrollTableCommand Class
MindFusion.Diagramming Namespace

Selection Class

Represents a set of items selected in the Diagram document.

Syntax

JavaScript Copy Code

```
// class
Selection.prototype = {}
```

Remarks

Several diagram items could be selected simultaneously. If more than one item is selected, the compound set of selected items is handled by an instance of the **Selection** class, returned by diagram's selection property. Items in the selection can be of any element type. The items property returns an array of all items, while more specific types are returned by links and nodes properties.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.Selection

See Also

Selection Members

Represents a set of items selected in the Diagram document.

The following tables list the members exposed by the Selection type.

Public Constructors

	Name	Description
	Selection	Initializes a new instance of the **Selection** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	allowMultipleSelection	Gets or sets a value indicating whether more than one items can be selected simultaneously.
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	items	Returns an array of all items in this selection.
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	links	Returns an array of all links in this selection.

	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	nodes	Returns an array of all nodes in this selection.
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	True if this is a top-level item in the diagram, or false if it is inside a container. (Inherited from DiagramItem.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	addItem	Adds an item to the **Selection**.
	allowDrag	DiagramItem.allowDrag override.
	Animate	Animates the item. (Inherited from DiagramItem.)
	clear	Removes all items from the **selection**.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	ContainsPoint	Overloaded. Determines whether this item contains the specified point. (Inherited from DiagramItem.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	Draws the item on the specified IGraphics surface. (Inherited from DiagramItem.)
	DrawHandles	Renders the adjustment handles of this item. (Inherited from DiagramItem.)
	DrawInRect	(Inherited from DiagramItem.)
	DrawShadow	Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramItem.)
	endDrag	DiagramItem.endDrag override.
	getBounds	Gets a rectangle that defines the position of the selection frame.
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetOutlinePoly	Gets a polygon approximating the outline of this object. (Inherited from DiagramItem.)
	GetRepaintRect	Gets the repaint region for this item, taking into account factors such as pen, selection handles and shadow. (Inherited from DiagramItem.)
	GetRotatedBounds	Gets the bounding rectangle of this item when it is rotated. (Inherited from DiagramItem.)
	GetToolTip	Gets the tooltip that should be displayed for this item at the specified point. (Inherited from DiagramItem.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramItem.)
	HitTestManipulators	Finds the **Manipulator** object located at the specified point. (Inherited from DiagramItem.)
	LabelFromPoint	Returns the ItemLabel located at specified point. (Inherited from DiagramItem.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)

	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	onRemove	DiagramItem.onRemove override.
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	removeItem	Removes an item from the **Diagram**.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveLocationState	DiagramItem.saveLocationState override.
	startDrag	DiagramItem.startDrag override.
	updateDrag	DiagramItem.updateDrag override.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	Invoked during interactive creation of items to test whether the operation can be completed in the current context. (Inherited from DiagramItem.)
	AllowModify	Invoked during interactive modification of items to test whether the operation can be completed in the current context. (Inherited from DiagramItem.)
	CancelModify	Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramItem.)
	CompleteCreate	Invoked when the item creation has been completed. (Inherited from DiagramItem.)
	CompleteModify	Invoked when the item modification has been completed. (Inherited from DiagramItem.)
	CreateProperties	Creates an item property set corresponding to the type of this diagram item. (Inherited from DiagramItem.)
	CreateState	Creates an item state corresponding to the type of this diagram item. (Inherited from DiagramItem.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the item is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramItem.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)

	LoadFromJson	Loads the item from a JSON object. (Inherited from DiagramItem.)
	LoadFromXml	Loads the item content from an XML element. (Inherited from DiagramItem.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	Called when the item is added to a Diagram, either by the user, programmatically or via undo / redo. (Inherited from DiagramItem.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	Called when the diagram's MeasureUnit changes. (Inherited from DiagramItem.)
	OnRemove	Called when an item is removed from a **Diagram**, either by the user, programmatically or via undo / redo. (Inherited from DiagramItem.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnZoomChanged	Called when the diagram's **ZoomFactor** changes. (Inherited from DiagramItem.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	Restores the properties of this diagram item. (Inherited from DiagramItem.)
	RestoreState	Restores the state of this diagram item. (Inherited from DiagramItem.)
	SaveProperties	Saves the properties of this diagram item. (Inherited from DiagramItem.)
	SaveState	Saves the state of this diagram item. (Inherited from DiagramItem.)
	SaveToJson	Saves the item as a JSON object. (Inherited from DiagramItem.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramItem.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpanded	Sets whether the item is in expanded or collapsed state. (Inherited from DiagramItem.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	StartCreate	Invoked once when the user starts creating a new item interactively. (Inherited from DiagramItem.)
	StartModify	Invoked when the user starts modifying this item interactively. (Inherited from DiagramItem.)
	UpdateCreate	Invoked during interactive creation of this item. (Inherited from DiagramItem.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from

		DiagramItem.)
	UpdateModify	Invoked during interactive modification of this item. (Inherited from DiagramItem.)

See Also

Selection Class
MindFusion.Diagramming Namespace

Selection Fields

Selection Constructor

Initializes a new instance of the Selection class.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
function Selection (parent)
```

Parameters

parent
The diagram where the items are selected.

See Also

Selection Class
MindFusion.Diagramming Namespace

Selection Methods

Selection.addItem Method

Adds an item to the Selection.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
function addItem (item)
```

Parameters

item
The item to add.

Remarks

This method adds a new item to the selection. It also deactivates the activeItem, if any, and makes the newly added item the active one.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.allowDrag Method

DiagramItem.allowDrag override.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript — Copy Code

```
function allowDrag (ist)
```

Parameters

ist
InteractionState. An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.clear Method

Removes all items from the selection.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript — Copy Code

```
function clear ()
```

Remarks

Call clear to deselect all items. The items are not deleted. The method also sets the diagram's activeItem to Null.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.endDrag Method

DiagramItem.endDrag override.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript — Copy Code

```
function endDrag (ist)
```

Parameters

ist
InteractionState. An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.getBounds Method

Gets a rectangle that defines the position of the selection frame.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
function getBounds ()
```

Return Value

Rect. The frame's bounding rectangle.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.onRemove Method

DiagramItem.onRemove override.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
function onRemove ()
```

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.removeItem Method

Removes an item from the Diagram.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
function removeItem (item)
```

Parameters

item
The item to add.

Remarks

Call this method to deselect an item. Note that the item is not deleted, but just removed from the selection.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.saveLocationState Method

DiagramItem.saveLocationState override.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
function saveLocationState ()
```

Return Value

Object. An object containing selection state.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.startDrag Method

DiagramItem.startDrag override.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
function startDrag ([ist])
```

Parameters

ist
Optional. An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.updateDrag Method

DiagramItem.updateDrag override.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
function updateDrag (ist)
```

Parameters

ist

An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection Properties

Selection.allowMultipleSelection Property

Gets or sets a value indicating whether more than one items can be selected simultaneously.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
get allowMultipleSelection() {}
set allowMultipleSelection(value) {}
```

Property Value

Boolean

true to let the user select multiple items, or **false** otherwise. The default value is **true**.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.items Property

Returns an array of all items in this selection.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
get items() {}
```

Property Value

Array

Array of DiagramItem objects.

Remarks

Do not modify this array directly, but call Selection methods such as addItem and removeItem, which also update item's state variables and redraw the diagram.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.links Property

Returns an array of all links in this selection.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
get links() {}
```

Property Value

DiagramLink

Array of DiagramLink objects.

Remarks

Do not modify this array directly, but call Selection methods such as addItem and removeItem, which also update item's state variables and redraw the diagram.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.nodes Property

Returns an array of all nodes in this selection.

Namespace: MindFusion.Diagramming
Assembly: Selection.js

Syntax

JavaScript Copy Code

```
get nodes() {}
```

Property Value

DiagramNode

Array of DiagramNode objects.

Remarks

Do not modify this array directly, but call Selection methods such as addItem and removeItem, which also update item's state variables and redraw the diagram.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

SelectionChangedEventArgs Class

Contains the arguments passed to handlers of the selectionChanged event.

Syntax

JavaScript Copy Code

```
// class
SelectionChangedEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.EventArgs
  MindFusion.Diagramming.SelectionChangedEventArgs

See Also

SelectionChangedEventArgs Members
MindFusion.Diagramming Namespace
Events Class

SelectionChangedEventArgs Members

Contains the arguments passed to handlers of the selectionChanged event.

The following tables list the members exposed by the SelectionChangedEventArgs type.

Public Constructors

	Name	Description
	SelectionChangedEventArgs	Initializes a new instance of the SelectionChangedEventArgs class.

Public Properties

	Name	Description
	handled	Gets or sets a value indicating whether the event has been handled. (Inherited from EventArgs.)
	newItems	Gets the selected items after the change.
	oldItems	Gets the selected items before the change.

See Also

SelectionChangedEventArgs Class
MindFusion.Diagramming Namespace

SelectionChangedEventArgs Constructor

Initializes a new instance of the SelectionChangedEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function SelectionChangedEventArgs (args)
```

Parameters

args

See Also

SelectionChangedEventArgs Class
MindFusion.Diagramming Namespace

SelectionChangedEventArgs Properties

SelectionChangedEventArgs.newItems Property

Gets the selected items after the change.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get newItems() {}
set newItems(value) {}
```

Property Value

Array

An array of items.

See Also

SelectionChangedEventArgs Members
SelectionChangedEventArgs Class
MindFusion.Diagramming Namespace

SelectionChangedEventArgs.oldItems Property

Gets the selected items before the change.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get oldItems() {}
set oldItems(value) {}
```

Property Value

Array

An array of items.

See Also

SelectionChangedEventArgs Members
SelectionChangedEventArgs Class
MindFusion.Diagramming Namespace

SelectionEventArgs Class

Contains the arguments passed to handlers of selection-related events.

Syntax

JavaScript — Copy Code

```
// class
SelectionEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.CancelEventArgs
  MindFusion.Diagramming.SelectionEventArgs

See Also

SelectionEventArgs Members
MindFusion.Diagramming Namespace
Events Class

SelectionEventArgs Members

Contains the arguments passed to handlers of selection-related events.

The following tables list the members exposed by the SelectionEventArgs type.

Public Constructors

	Name	Description
	SelectionEventArgs	Initializes a new instance of the SelectionEventArgs class.

Public Properties

	Name	Description
	adjustmentHandle	Gets the adjustment handle being moved by the user.
	cancel	Gets or sets a value indicating whether to allow the current operation. (Inherited from CancelEventArgs.)
	handled	Gets or sets a value indicating whether the event has been handled. (Inherited from EventArgs.)
	mousePosition	Gets the current mouse pointer position.

Public Methods

	Name	Description
	cancelDrag	Immediately cancels the current selection modification.

See Also

SelectionEventArgs Class
MindFusion.Diagramming Namespace

SelectionEventArgs Constructor

Initializes a new instance of the SelectionEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function SelectionEventArgs (args)
```

Parameters

args

See Also

SelectionEventArgs Class
MindFusion.Diagramming Namespace

SelectionEventArgs Methods

SelectionEventArgs.cancelDrag Method

Immediately cancels the current selection modification.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function cancelDrag ()
```

See Also

SelectionEventArgs Members
SelectionEventArgs Class
MindFusion.Diagramming Namespace

SelectionEventArgs Properties

SelectionEventArgs.adjustmentHandle Property

Gets the adjustment handle being moved by the user.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get adjustmentHandle() {}
```

Property Value

AdjustmentHandles

A member of the AdjustmentHandles enumeration.

See Also

SelectionEventArgs Members
SelectionEventArgs Class
MindFusion.Diagramming Namespace

SelectionEventArgs.mousePosition Property

Gets the current mouse pointer position.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript — Copy Code

```
get mousePosition() {}
```

Property Value

Point

A Point instance containing mouse pointer coordinates.

See Also

SelectionEventArgs Members
SelectionEventArgs Class
MindFusion.Diagramming Namespace

SelectOnlyBehavior Class

Allows only selection of existing items. Modifying them or drawing new ones is disabled.

Syntax

JavaScript — Copy Code

```
// class
SelectOnlyBehavior.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.BehaviorBase
MindFusion.Diagramming.SelectOnlyBehavior

See Also

SelectOnlyBehavior Members
MindFusion.Diagramming Namespace

SelectOnlyBehavior Members

Allows only selection of existing items. Modifying them or drawing new ones is disabled.

The following tables list the members exposed by the SelectOnlyBehavior type.

Public Constructors

	Name	Description
	SelectOnlyBehavior	Overloaded.

Public Properties

	Name	Description
	diagram	Gets the target diagram of this behavior object. (Inherited from BehaviorBase.)

Public Methods

	Name	Description
	createController	BehaviorBase.createController override.

See Also

SelectOnlyBehavior Class
MindFusion.Diagramming Namespace

SelectOnlyBehavior Constructor

Overload List

Name	Description
SelectOnlyBehavior (Diagram)	Initializes a new instance of the SelectOnlyBehavior class.
SelectOnlyBehavior (DiagramView)	Initializes a new instance of the SelectOnlyBehavior class.

See Also

SelectOnlyBehavior Class
MindFusion.Diagramming Namespace

SelectOnlyBehavior Constructor (Diagram)

Initializes a new instance of the SelectOnlyBehavior class.

Namespace: MindFusion.Diagramming
Assembly: SelectOnlyBehavior.js

Syntax

JavaScript Copy Code

```
function SelectOnlyBehavior (parent)
```

Parameters

parent
The Diagram this behavior will be associated with.

See Also

SelectOnlyBehavior Constructor Overload List
SelectOnlyBehavior Class
MindFusion.Diagramming Namespace

SelectOnlyBehavior Constructor (DiagramView)

Initializes a new instance of the SelectOnlyBehavior class.

Namespace: MindFusion.Diagramming
Assembly: SelectOnlyBehavior.js

Syntax

JavaScript Copy Code

```
function SelectOnlyBehavior (parent)
```

Parameters

parent
DiagramView. The DiagramView this behavior will be associated with.

See Also

SelectOnlyBehavior Constructor Overload List
SelectOnlyBehavior Class
MindFusion.Diagramming Namespace

SelectOnlyBehavior Methods

SelectOnlyBehavior.createController Method

BehaviorBase.createController override.

Namespace: MindFusion.Diagramming
Assembly: SelectOnlyBehavior.js

Syntax

JavaScript Copy Code

```
function createController (state)
```

Parameters

state
Object. An object representing current context and input event.

Return Value

SinglePointerController. An instance of SinglePointerController -derived class.

See Also

SelectOnlyBehavior Members
SelectOnlyBehavior Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs Class

Contains the arguments passed to the serializeTag and deserializeTag event handlers.

Syntax

JavaScript Copy Code

```
// class
SerializeTagEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.EventArgs
MindFusion.Diagramming.SerializeTagEventArgs

See Also

SerializeTagEventArgs Members
MindFusion.Diagramming Namespace

SerializeTagEventArgs Members

Contains the arguments passed to the serializeTag and deserializeTag event handlers.

The following tables list the members exposed by the SerializeTagEventArgs type.

Public Constructors

	Name	Description
	SerializeTagEventArgs	Initializes a new instance of the SerializeTagEventArgs class.

Public Properties

	Name	Description
	context	Gets an XmlPersistContext object providing helper read and write methods for various value types.
	element	Gets the XML element where the tag value should be serialized to or deserialized from.
	jsonObject	Gets or sets the object representing the tag.
	object	Gets the object whose property is being serialized or deserialized.
	propertyName	Gets the name of the property being serialized.
	tag	Gets or sets the tag value being serialized or deserialized.

See Also

SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs Constructor

Initializes a new instance of the SerializeTagEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function SerializeTagEventArgs ()
```

See Also

SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs Properties

SerializeTagEventArgs.context Property

Gets an XmlPersistContext object providing helper read and write methods for various value types.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

```
JavaScript
get context() {}
```

Property Value

XmlPersistContext

An instance of the XmlPersistContext class.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs.element Property

Gets the XML element where the tag value should be serialized to or deserialized from.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

```
JavaScript
get element() {}
```

Property Value

Element

The XML element.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs.jsonObject Property

Gets or sets the object representing the tag.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

```
JavaScript
get jsonObject() {}
set jsonObject(value) {}
```

Property Value

Object

The object represeting the tag.

Remarks

Used in JSON serialization.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs.object Property

Gets the object whose property is being serialized or deserialized.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get object() {}
```

Property Value

Object

An object whose custom data is being serialized.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs.propertyName Property

Gets the name of the property being serialized.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get propertyName() {}
```

Property Value

String

A string containing property name.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs.tag Property

Gets or sets the tag value being serialized or deserialized.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get tag() {}
set tag(value) {}
```

Property Value

Object

An object containing custom data.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

ShadowsStyle Enumeration

Enumerates possible shadows-drawing styles.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
ShadowsStyle = {}
```

Members

	Member name	Description
	None	No shadows are drawn.
	OneLevel	All shadows are drawn at the lowest z-level, and appear to be on one level.
	ZOrder	Every shadows is drawn just below its object, possibly covering other objects.

Remarks

Members of this enumeration can be assigned to the shadowsStyle property of Diagram.

See Also

MindFusion.Diagramming Namespace

Shape Class

An instance of this class represents a shape defined through a series of arc, line and Bezier segments. The segments can be a part of the shape outline, in which case they define the part of the shape is filled, used for hit-testing and for aligning links to the node edges. Segments can also be added as decoration elements, in which case they only affect the appearance of a ShapeNode.

Syntax

JavaScript Copy Code

```
// class
Shape.prototype = {}
```

Remarks

The **Shape** class can also be used as a component in composite nodes. When its **isOutline** attribute is set, the shape will control CompositeNode's geometry, defining hit test and clip area, and link alignment points along node's border. If **isOutline** is disabled, the shape will serve mostly as a decorative element. The code below shows a sample fragment specifying shape component in template:

JavaScript Copy Code

```
{
    component: "Shape",
    id: "Cloud",
    autoProperty: true,
    name: "OutlineShape",
```

```
    pen: "gray",
    brush: "Transparent",
    isOutline: true
},
```

Example

Custom shapes can be defined using SVG Path syntax. The following code creates custom shape geoometry and assigns it to a ShapeNode:

JavaScript Copy Code

```
var shape = new MindFusion.Diagramming.Shape({
   id: "DoubleEllipse",
   outline: "M100,50 A50,50,0,1,1,0,50 A50,50,0,1,1,100,50",
   decoration: "M98,50 A48,48,0,1,1,2,50 A48,48,0,1,1,98,50",
   fillMode: "nonzero"});

node.shape = shape;
```

Inheritance Hierarchy

MindFusion.Diagramming.ComponentBase
  MindFusion.Diagramming.Shape

See Also

Shape Members
MindFusion.Diagramming Namespace

Shape Members

An instance of this class represents a shape defined through a series of arc, line and Bezier segments. The segments can be a part of the shape outline, in which case they define the part of the shape is filled, used for hit-testing and for aligning links to the node edges. Segments can also be added as decoration elements, in which case they only affect the appearance of a ShapeNode.

The following tables list the members exposed by the Shape type.

Public Constructors

	Name	Description
	Shape	Initializes a new instance of the Shape class.

Public Properties

	Name	Description
	anchorPattern	Gets or sets the default anchor pattern for nodes that use this shape.
	brush	Gets or sets the brush used to paint the interior of this shape.
	pen	Gets or sets the pen used to stroke the frame of this shape.
	strokeThickness	Gets or sets the thickness of the shape's frame.

Public Methods

	Name	Description
S	component	Creates a shape component.
S	fromId	Returns a reference to the **Shape** with the specified string identifier.
	getId	Gets the id of this **Shape**.
	isElliptic	Gets a value indicating whether this shape is elliptic.

	loadFromXml	Loads the shape content from an XML element.

See Also

Shape Class
MindFusion.Diagramming Namespace

Shape Constructor

Initializes a new instance of the Shape class.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript Copy Code

```
function Shape ([params])
```

Parameters

params.outline
Optional.
String. A string defining the shape's outline.

params.decoration
Optional.
String. A string defining the shape's decoration.

params.shapeDecorations
Optional.
Array. An array containing shapeDecoration objects defining additional shape decoration.

params.id
Optional.
String. The string identifier of the shape.

params.fillMode
Optional.
String. A string specifying the fill mode;

params.outlinePen
Optional.
String. A string specifying the stroke of the shape's outline.

params.outlineBrush
Optional.
String. A string specifying the fill of the shape's outline.

params.decorationPen
Optional.
String. A string specifying the stroke of the shape's decoration.

params.decorationBrush
Optional.
String. A string specifying the fill of the shape's decoration.

See Also

Shape Class
MindFusion.Diagramming Namespace

Shape Methods

Shape.component Method

Creates a shape component.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript — Copy Code

```
function component (id, [isOutline, [node]])
```

Parameters

id
String. The identifier of the shape.

isOutline
Optional.
Boolean. A value indicating whether this shape defines the outline of its host node.

node
Optional.
CompositeNode. The host node of this component.

Return Value

Shape. A reference to a Shape component with the specified identifier.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.fromId Method

Returns a reference to the Shape with the specified string identifier.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript — Copy Code

```
function fromId (id)
```

Parameters

id
The identifier of the Shape of interest.

Return Value

A reference to the **Shape** with the specified identifier or **null** if there is no such shape.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.getId Method

Gets the id of this Shape.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript — Copy Code

```
function getId ()
```

Return Value

The string identifier.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.isElliptic Method

Gets a value indicating whether this shape is elliptic.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript Copy Code

```
function isElliptic ()
```

Return Value

Boolean. True if the shape is elliptic, or false otherwise.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.loadFromXml Method

Loads the shape content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement

Element. An XML DOM element that contains the the shape's data.

context

XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape Properties

Shape.anchorPattern Property

Gets or sets the default anchor pattern for nodes that use this shape.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript — Copy Code

```
get anchorPattern() {}
set anchorPattern(value) {}
```

Property Value

AnchorPattern

An AnchorPattern instance. The default value is **null**.

Remarks

If the anchorPattern of a ShapeNode is set, it overrides the one associated with the Shape definition.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.brush Property

Gets or sets the brush used to paint the interior of this shape.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript — Copy Code

```
get brush() {}
set brush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

Remarks

Used to paint Shape components of CompositeNode.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.pen Property

Gets or sets the pen used to stroke the frame of this shape.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript Copy Code

```
get pen() {}
set pen(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

Remarks

Used to paint Shape components of CompositeNode.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.strokeThickness Property

Gets or sets the thickness of the shape's frame.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript Copy Code

```
get strokeThickness() {}
set strokeThickness(value) {}
```

Property Value

Number

A number specifying stroke thickness.

Remarks

Used to paint Shape components of CompositeNode.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

ShapeDecoration Class

An instance of this class represents a child decoration figure of shape's geometry.

Syntax

JavaScript Copy Code

```
// class
ShapeDecoration.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.ShapeDecoration

See Also

ShapeDecoration Members

An instance of this class represents a child decoration figure of shape's geometry.

The following tables list the members exposed by the ShapeDecoration type.

Public Constructors

	Name	Description
	ShapeDecoration	Initializes a new instance of the **ShapeDecoration** class.

See Also

ShapeDecoration Class
MindFusion.Diagramming Namespace

ShapeDecoration Constructor

Initializes a new instance of the ShapeDecoration class.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript Copy Code

```
function ShapeDecoration ([elements, [fillColor, [strokeColor, [strokeThickness,
[strokeDashStyle]]]]])
```

Parameters

elements
Optional.
Array. An array of ShapeElement objects.

fillColor
Optional.
String. Optional fill color.

strokeColor
Optional.
String. Optional stroke color.

strokeThickness
Optional.
Number. Optional stroke thickness.

strokeDashStyle
Optional.
DashStyle. Optional dash style.

See Also

ShapeDecoration Class
MindFusion.Diagramming Namespace

ShapeElement Class

An instance of this class represents an element of shape's geometry, such as line segment, arc or Bezier curve.

Syntax

JavaScript Copy Code

```
// class
ShapeElement.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.ShapeElement

See Also

ShapeElement Members
MindFusion.Diagramming Namespace

ShapeElement Members

An instance of this class represents an element of shape's geometry, such as line segment, arc or Bezier curve.

The following tables list the members exposed by the ShapeElement type.

Public Constructors

	Name	Description
	ShapeElement	Initializes a new instance of the **ShapeElement** class.

See Also

ShapeElement Class
MindFusion.Diagramming Namespace

ShapeElement Constructor

Initializes a new instance of the ShapeElement class.

Namespace: MindFusion.Diagramming
Assembly: Shape.js

Syntax

JavaScript Copy Code

```
function ShapeElement ([func, [args, [strokeColor, [strokeThickness, [strokeDashStyle]]]]])
```

Parameters

func
Optional.
String. One of the moveTo, lineTo, arc, bezierCurveTo, roundRect functions.

args
Optional.
Array. Coordinates to pass to the drawing function.

strokeColor
Optional.
String. Optional color to apply to the element.

strokeThickness
Optional.
Number. Optional thickness to apply to the element.

strokeDashStyle
Optional.
DashStyle. Optional dash style to apply to the element.

See Also

ShapeElement Class
MindFusion.Diagramming Namespace

ShapeLibrary Class

The **ShapeLibrary** class represents a collection of shape definitions stored together.

Syntax

JavaScript Copy Code

```
// class
ShapeLibrary.prototype = {}
```

Remarks

Shape libraries can be created using the ShapeDesigner tool that comes with MindFusion.Diagramming. Use the loadFromXml method to load a shape library from a file. The loaded shapes can be accessed through the Shapes property.

Inheritance Hierarchy

MindFusion.Diagramming.ShapeLibrary

See Also

ShapeLibrary Members
MindFusion.Diagramming Namespace

ShapeLibrary Members

The ShapeLibrary class represents a collection of shape definitions stored together.

The following tables list the members exposed by the **ShapeLibrary** type.

Public Constructors

	Name	Description
	ShapeLibrary	Initializes a new instance of the ShapeLibrary class.

Public Fields

	Name	Description
	shapes	A list containing all the shapes.

Public Methods

	Name	Description
	getShapes	Gets the list with all shapes contained in the library.
	loadFromXml	Overloaded.

See Also

ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary Fields

ShapeLibrary.shapes Field

A list containing all the shapes.

Namespace: MindFusion.Diagramming
Assembly: ShapeLibrary.js

Syntax

JavaScript Copy Code

```
var shapes
```

See Also

ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary Constructor

Initializes a new instance of the ShapeLibrary class.

Namespace: MindFusion.Diagramming
Assembly: ShapeLibrary.js

Syntax

JavaScript Copy Code

```
function ShapeLibrary (shapes)
```

Parameters

shapes
Array. A list of MindFusion.Diagramming.Shape objects to initialize the library with.

See Also

ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary Methods

ShapeLibrary.getShapes Method

Gets the list with all shapes contained in the library.

Namespace: MindFusion.Diagramming
Assembly: ShapeLibrary.js

Syntax

JavaScript Copy Code

```
function getShapes ()
```

Return Value

Array. A list with all the MindFusion.Diagramming.Shape objects contained in the library.

See Also

ShapeLibrary Members
ShapeLibrary Class

ShapeLibrary.loadFromXml Method

Overload List

Name	Description
ShapeLibrary.loadFromXml (String, function, function)	Loads the library from the specified XML document.
ShapeLibrary.loadFromXml (String, Function, Function)	Loads the library from the specified XML document.

See Also

ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary.loadFromXml Method (String, function, function)

Loads the library from the specified XML document.

Namespace: MindFusion.Diagramming
Assembly: ShapeLibrary.js

Syntax

JavaScript Copy Code

```
function loadFromXml (fileUrl, [onLoad, [onError]])
```

Parameters

fileUrl
String. The URL of an XML file where the data should be read from.

onLoad
Optional.
function. A callback that should be invoked if the file is loaded successfully.

onError
Optional.
function. A callback that should be invoked if the file could not be downloaded.

See Also

loadFromXml Method Overload List
ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary.loadFromXml Method (String, Function, Function)

Loads the library from the specified XML document.

Namespace: MindFusion.Diagramming
Assembly: ShapeLibrary.js

Syntax

JavaScript Copy Code

```
function loadFromXml (fileUrl, [onLoad, [onError]])
```

Parameters

fileUrl
The URL of an XML file where the data should be read from.

onLoad
Optional.
A callback that should be invoked if the file is loaded successfully.

onError
Optional.
A callback that should be invoked if the file could not be downloaded.

See Also

loadFromXml Method Overload List
ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeNode Class

ShapeNode instances are diagram nodes that represent geometric shapes.

Syntax

JavaScript Copy Code

```
// class
ShapeNode.prototype = {}
```

Remarks

ShapeNode instance might represent nodes in graphs, tree, process diagrams, and so on. Node's geometry is specified via its shape property. The default shape of new nodes is specified by diagram's defaultShape property. Shapes can be rotated, either interactively using the rotation handle above the node, or programmatically using the rotationAngle property. Shape nodes can display images and text.

Assign an image to be painted inside a node by setting image, imageLocation or imageContent properties. Image position and alignment are set via the imagePadding and imageAlign properties.

Text inside a shape node is represented by the text property. Its appearance can be customized using textColor and font properties. To get even richer text formatting, set enableStyledText to **true** and embed HTML-like formatting tags in the raw text. To make a node big enough to display its text without clipping, call resizeToFitText.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.ShapeNode

See Also

ShapeNode Members
MindFusion.Diagramming Namespace

ShapeNode Members

ShapeNode instances are diagram nodes that represent geometric shapes.

The following tables list the members exposed by the **ShapeNode** type.

Public Constructors

	Name	Description

	ShapeNode	Initializes a new instance of the **ShapeNode** class with default values supplied form the specified Diagram.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	allowFlip	Gets or sets whether the node's shape flips when users drag an edge over its opposite edge.
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	anchorPattern	DiagramNode.anchorPattern override.
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	clipText	Gets or sets a value indicating whether to clip the text assigned to this node to fit into the shape's geometry.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	cssElementName	Gets the name of the CSS element associated with this item.
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	enableStyledText	Gets or sets a value indicating whether styled text rendering is enabled.
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)

	flipX	Gets or sets a value indicating whether the node's shape is fliped horizontally.
	flipY	Gets or sets a value indicating whether the node's shape is fliped vertically.
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	image	Gets or sets the image displayed by this node.
	imageAlign	Gets or sets the alignment of node's image.
	imageContent	Gets or sets a Base64-encoded image to display inside this node.
	imageLocation	Gets or sets URL of the image to display inside this node.
	imagePadding	Gets or sets the padding distance between node borders and image.
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	polygonalTextLayout	Gets or sets a value indicating whether text is wrapped to fit the node's shape.
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	rotateImage	Gets or sets a value indicating whether image gets rotated along with the node.

	rotateText	Gets or sets a value indicating whether text gets rotated along with the node.
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	shape	Gets or sets the geometric shape displayed by this node.
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	transparent	Gets or sets a value indicating whether the node's geometry is visible.
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description

	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	clone	DiagramNode.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	containsPoint	Checks if the **ShapeNode** contains the specified point.
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	fromJson	DiagramNode.fromJson override.
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)

	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	getOutline	Gets an array of points that define the outline of this node shape.
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	loadFromXml	DiagramNode.LoadFromXml override. Loads the node's content from an XML element.
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)

	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	resizeToFitText	Makes the shape node big enough to display its text without clipping.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	restoreLocationState	DiagramItem.restoreLocationState override.
	restoreState	DiagramNode.restoreState override.
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveLocationState	DiagramItem.saveLocationState override.
	saveState	DiagramNode.saveState override.
	saveToXml	Saves the node's content into an XML element.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	toJson	DiagramNode.toJson override.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	updateDrag	DiagramItem.updateDrag override.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)

	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object. (Inherited from DiagramNode.)
	LoadFromXml	DiagramItem.LoadFromXml override. (Inherited from DiagramNode.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)

	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from DiagramNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode Constructor

Initializes a new instance of the ShapeNode class with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
function ShapeNode ([parent])
```

Parameters

parent
Optional. The Diagram from which to obtain default values for the node properties.

Remarks

If parent is not specified, this constructor initializes the ShapeNode properties with the values specified in the respective default-value properties of the Diagram object.

If parent is specified, the constructor copies the ShapeNode property values from the specified prototype node.

See Also

ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode Methods

ShapeNode.clone Method

DiagramNode.clone override.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
function clone ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.containsPoint Method

Checks if the ShapeNode contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
function containsPoint (point)
```

Parameters

point
The Point to check.

Return Value

true if the **Point** is within the bounds of this node; otherwise, false.

Remarks

This method is used during hit-testing to see if the mouse is inside the item.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.fromJson Method

DiagramNode.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
function fromJson ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.getOutline Method

Gets an array of points that define the outline of this node shape.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
function getOutline ()
```

Return Value

Array. An array with the points.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.loadFromXml Method

DiagramNode.LoadFromXml override. Loads the node's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context

XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.resizeToFitText Method

Makes the shape node big enough to display its text without clipping.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
function resizeToFitText (fit)
```

Parameters

fit

One of the FitSize enumeration values.

Return Value

true if the node is resized successfully; otherwise, **false**.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
text Property
font Property

ShapeNode.restoreLocationState Method

DiagramItem.restoreLocationState override.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
function restoreLocationState (ist)
```

Parameters

ist

ShapeNodeState. A ShapeNodeState instance.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.restoreState Method

DiagramNode.restoreState override.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

```
JavaScript                                                    Copy Code
function restoreState ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.saveLocationState Method

DiagramItem.saveLocationState override.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

```
JavaScript                                                    Copy Code
function saveLocationState ()
```

Return Value

ShapeNodeState. A ShapeNodeState instance.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.saveState Method

DiagramNode.saveState override.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

```
JavaScript                                                    Copy Code
function saveState ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.saveToXml Method

Saves the node's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement

Element. An XML DOM element that will contain the item's data.

context

XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.toJson Method

DiagramNode.toJson override.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.updateDrag Method

DiagramItem.updateDrag override.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
function updateDrag (ist)
```

Parameters

ist

InteractionState. An InteractionState object that encapsulates the current state of user interaction with the control.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode Properties

ShapeNode.allowFlip Property

Gets or sets whether the node's shape flips when users drag an edge over its opposite edge.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript — Copy Code

```
get allowFlip() {}
set allowFlip(value) {}
```

Property Value

Boolean

true to flip the shape, or **false** otherwise.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.anchorPattern Property

DiagramNode.anchorPattern override.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript — Copy Code

```
get anchorPattern() {}
set anchorPattern(value) {}
```

Property Value

AnchorPattern

An AnchorPattern object that defines a set of anchor points.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.clipText Property

Gets or sets a value indicating whether to clip the text assigned to this node to fit into the shape's geometry.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript — Copy Code

```
get clipText() {}
set clipText(value) {}
```

Property Value

Boolean

true to clip the node's text, or **false** otherwise. The default value is **false**.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
text Property
rotateText Property

ShapeNode.cssElementName Property

Gets the name of the CSS element associated with this item.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get cssElementName() {}
set cssElementName(value) {}
```

Property Value

String

A string containing the CSS element name.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.enableStyledText Property

Gets or sets a value indicating whether styled text rendering is enabled.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get enableStyledText() {}
set enableStyledText(value) {}
```

Property Value

Boolean

true to parse the node's text in order to find formatting tags, or **false** otherwise.

Remarks

Styled text allows applying various attributes to the characters it contains. Attributes are specified using HTML-like formatting tags embedded in the raw text. At this time the component supports the following formatting tags:

- <b> specifies bold text
- <i> specifies italic text
- <u> specifies underlined text
- <color=value> specifies text color
- <sub>specifies subscript
- <sup>specifies superscript
-
 indicates line break

- <a=url> renders specified text as a hyperlink and raises hyperlinkClicked event when link has been clicked
- <size=value>specifies font size

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
text Property
polygonalTextLayout Property
font Property
textColor Property

ShapeNode.flipX Property

Gets or sets a value indicating whether the node's shape is fliped horizontally.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get flipX() {}
set flipX(value) {}
```

Property Value

Boolean

true if the shape is flipped horizontally, or **false** otherwise.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.flipY Property

Gets or sets a value indicating whether the node's shape is fliped vertically.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get flipY() {}
set flipY(value) {}
```

Property Value

Boolean

true if the shape is flipped vertically, or **false** otherwise.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.image Property

Gets or sets the image displayed by this node.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get image() {}
set image(value) {}
```

Property Value

HTMLImageElement

An image object to render as node's content.

Remarks

This property lets you display an image inside the node. Set the imageAlign property to specify how the image is aligned inside shape's area. By default, the image is displayed in the center of the bounding rectangle of the node.

If a node should display multiple images, you could add and arrange them using NodeLabel objects.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.imageAlign Property

Gets or sets the alignment of node's image.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get imageAlign() {}
set imageAlign(value) {}
```

Property Value

ImageAlign

A member of the ImageAlign enumeration.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.imageContent Property

Gets or sets a Base64-encoded image to display inside this node.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get imageContent() {}
set imageContent(value) {}
```

Property Value

String

A Base64-encoded representation of an image.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.imageLocation Property

Gets or sets URL of the image to display inside this node.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get imageLocation() {}
set imageLocation(value) {}
```

Property Value

String

A URL string that identifies the image to be loaded.

Remarks

This property lets you display an image inside the node. Set the imageAlign property to specify how the image is aligned inside shape's area. By default, the image is displayed in the center of the bounding rectangle of the node.

If a node should display multiple images, you could add and arrange them using NodeLabel objects.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.imagePadding Property

Gets or sets the padding distance between node borders and image.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get imagePadding() {}
set imagePadding(value) {}
```

Property Value

Thickness

A Thickness instance specifying image padding.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.polygonalTextLayout Property

Gets or sets a value indicating whether text is wrapped to fit the node's shape.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get polygonalTextLayout() {}
set polygonalTextLayout(value) {}
```

Property Value

Boolean

true to wrap the text according to node's geometry, or **false** otherwise.

Remarks

The formatting algorithm is designed to lay out text inside polygonal outlines. A polygonal approximation is created for outlines that contain curved segments, so text can also be laid out relatively to Bézier or spline segments.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
text Property
enableStyledText Property

ShapeNode.rotateImage Property

Gets or sets a value indicating whether image gets rotated along with the node.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get rotateImage() {}
set rotateImage(value) {}
```

Property Value

Boolean

true to rotate the image, or **false** otherwise.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.rotateText Property

Gets or sets a value indicating whether text gets rotated along with the node.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get rotateText() {}
set rotateText(value) {}
```

Property Value

Boolean

true to rotate the text, or **false** to keep it horizontal.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
text Property
clipText Property

ShapeNode.shape Property

Gets or sets the geometric shape displayed by this node.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get shape() {}
set shape(value) {}
```

Property Value

Shape | *String*

The Shape of this node.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
imageLocation Property
allowFlip Property
rotationAngle Property
cornerRadius Property

ShapeNode.transparent Property

Gets or sets a value indicating whether the node's geometry is visible.

Namespace: MindFusion.Diagramming
Assembly: ShapeNode.js

Syntax

JavaScript Copy Code

```
get transparent() {}
set transparent(value) {}
```

Property Value

Boolean

true if node's geometry is hidden, or **false** otherwise.

Remarks

If a node is transparent, only its image and text are drawn. Shape, filling and frames are not painted.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

Shapes Class

Provides alternative methods of access to the predefined shapes.

Syntax

JavaScript Copy Code

```
// class
Shapes.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Shapes

See Also

Shapes Members
MindFusion.Diagramming Namespace

Shapes Members

Provides alternative methods of access to the predefined shapes.

The following tables list the members exposed by the Shapes type.

Public Methods

	Name	Description
	Actor	Gets the Actor predefined shape.
	Alternative	Gets the Alternative predefined shape.
	Arrow	Gets the Arrow predefined shape.
	Arrow1	Gets the Arrow1 predefined shape.
	Arrow2	Gets the Arrow2 predefined shape.
	Arrow3	Gets the Arrow3 predefined shape.
	Arrow4	Gets the Arrow4 predefined shape.
	Arrow5	Gets the Arrow5 predefined shape.
	Arrow6	Gets the Arrow6 predefined shape.
	Arrow7	Gets the Arrow7 predefined shape.
	Arrow8	Gets the Arrow8 predefined shape.
	Arrow9	Gets the Arrow9 predefined shape.
	BackSlash	Gets the BackSlash predefined shape.
	BeginLoop	Gets the BeginLoop predefined shape.
	BowArrow	Gets the BowArrow predefined shape.

	BpmnComplex	Gets the BpmnComplex predefined shape.
	BpmnDataBasedXor	Gets the BpmnDataBasedXor predefined shape.
	BpmnEndCancel	Gets the BpmnEndCancel predefined shape.
	BpmnEndCompensation	Gets the BpmnEndCompensation predefined shape.
	BpmnEndError	Gets the BpmnEndError predefined shape.
	BpmnEndLink	Gets the BpmnEndLink predefined shape.
	BpmnEndMessage	Gets the BpmnEndMessage predefined shape.
	BpmnEndMultiple	Gets the BpmnEndMultiple predefined shape.
	BpmnEndTerminate	Gets the BpmnEndTerminate predefined shape.
	BpmnEventBasedXor	Gets the BpmnEventBasedXor predefined shape.
	BpmnInclusive	Gets the BpmnInclusive predefined shape.
	BpmnIntermediateCancel	Gets the BpmnIntermediateCancel predefined shape.
	BpmnIntermediateCompensation	Gets the BpmnIntermediateCompensation predefined shape.
	BpmnIntermediateError	Gets the BpmnIntermediateError predefined shape.
	BpmnIntermediateLink	Gets the BpmnIntermediateLink predefined shape.
	BpmnIntermediateMessage	Gets the BpmnIntermediateMessage predefined shape.
	BpmnIntermediateMultiple	Gets the BpmnIntermediateMultiple predefined shape.
	BpmnIntermediateRule	Gets the BpmnIntermediateRule predefined shape.
	BpmnIntermediateTimer	Gets the BpmnIntermediateTimer predefined shape.
	BpmnParallelFork	Gets the BpmnParallelFork predefined shape.
	BpmnStartLink	Gets the BpmnStartLink predefined shape.
	BpmnStartMessage	Gets the BpmnStartMessage predefined shape.
	BpmnStartMultiple	Gets the BpmnStartMultiple predefined shape.
	BpmnStartRule	Gets the BpmnStartRule predefined shape.
	BpmnStartTimer	Gets the BpmnStartTimer predefined shape.
	Circle	Gets the Circle predefined shape.
	Cloud	Gets the Cloud predefined shape.
	Collate	Gets the Collate predefined shape.
	ConeDown	Gets the ConeDown predefined shape.
	ConeUp	Gets the ConeUp predefined shape.
	ConnectedIssues	Gets the ConnectedIssues predefined shape.
	CreateRequest	Gets the CreateRequest predefined shape.
	Cross	Gets the Cross predefined shape.
	Cube	Gets the Cube predefined shape.
	Cylinder	Gets the Cylinder predefined shape.
	Database	Gets the Database predefined shape.
	DataTransmition	Gets the DataTransmition predefined shape.

	DDelay	Gets the DDelay predefined shape.
	Decagon	Gets the Decagon predefined shape.
	Decision	Gets the Decision predefined shape.
	Decision2	Gets the Decision2 predefined shape.
	DefaultFlow	Gets the DefaultFlow predefined shape.
	Delay	Gets the Delay predefined shape.
	DInDelay	Gets the DInDelay predefined shape.
	DirectAccessStorage	Gets the DirectAccessStorage predefined shape.
	DiskStorage	Gets the DiskStorage predefined shape.
	Display	Gets the Display predefined shape.
	DividedEvent	Gets the DividedEvent predefined shape.
	DividedProcess	Gets the DividedProcess predefined shape.
	Document	Gets the Document predefined shape.
	Donut	Gets the Donut predefined shape.
	DoubleArrow	Gets the DoubleArrow predefined shape.
	DOutDelay	Gets the DOutDelay predefined shape.
	Ellipse	Gets the Ellipse predefined shape.
	EndLoop	Gets the EndLoop predefined shape.
	ExternalOrganization	Gets the ExternalOrganization predefined shape.
	ExternalProcess	Gets the ExternalProcess predefined shape.
	File	Gets the File predefined shape.
	FramedRectangle	Gets the FramedRectangle predefined shape.
	Gate	Gets the Gate predefined shape.
	Heart	Gets the Heart predefined shape.
	Heptagon	Gets the Heptagon predefined shape.
	Hourglass	Gets the Hourglass predefined shape.
	Input	Gets the Input predefined shape.
	InternalStorage	Gets the InternalStorage predefined shape.
	Interrupt	Gets the Interrupt predefined shape.
	Interrupt2	Gets the Interrupt2 predefined shape.
	IsoProcess	Gets the IsoProcess predefined shape.
	Junction	Gets the Junction predefined shape.
	Lightning	Gets the Lightning predefined shape.
	LinedDocument	Gets the LinedDocument predefined shape.
	ManualOperation	Gets the ManualOperation predefined shape.
	Merge	Gets the Merge predefined shape.
	MessageFromUser	Gets the MessageFromUser predefined shape.

	MessageToUser	Gets the MessageToUser predefined shape.
	Microform	Gets the Microform predefined shape.
	MicroformProcessing	Gets the MicroformProcessing predefined shape.
	MicroformRecording	Gets the MicroformRecording predefined shape.
	MultiDocument	Gets the MultiDocument predefined shape.
	MultiProc	Gets the MultiProc predefined shape.
	Octagon	Gets the Octagon predefined shape.
	OfflineStorage	Gets the OfflineStorage predefined shape.
	OffpageConnection	Gets the OffpageConnection predefined shape.
	OffpageReference	Gets the OffpageReference predefined shape.
	Or	Gets the Or predefined shape.
	Pentagon	Gets the Pentagon predefined shape.
	PentagonArrow	Gets the PentagonArrow predefined shape.
	Plaque	Gets the Plaque predefined shape.
	PointerArrow	Gets the PointerArrow predefined shape.
	PrimitiveFromCall	Gets the PrimitiveFromCall predefined shape.
	PrimitiveToCall	Gets the PrimitiveToCall predefined shape.
	Procedure	Gets the Procedure predefined shape.
	Process	Gets the Process predefined shape.
	Prysm	Gets the Prysm predefined shape.
	PunchedCard	Gets the PunchedCard predefined shape.
	PyramidDown	Gets the PyramidDown predefined shape.
	PyramidUp	Gets the PyramidUp predefined shape.
	Quill	Gets the Quill predefined shape.
	Rectangle	Gets the Rectangle predefined shape.
	Reversed	Gets the Reversed predefined shape.
	RevTriangle	Gets the RevTriangle predefined shape.
	RevWithCirc	Gets the RevWithCirc predefined shape.
	RevWithLine	Gets the RevWithLine predefined shape.
	Rhombus	Gets the Rhombus predefined shape.
	RightTriangle	Gets the RightTriangle predefined shape.
	RoundRect	Gets the RoundRect predefined shape.
	RSave	Gets the RSave predefined shape.
	Save	Gets the Save predefined shape.
	Slash	Gets the Slash predefined shape.
	Sort	Gets the Sort predefined shape.
	Sphere	Gets the Sphere predefined shape.

	Star	Gets the Star predefined shape.
	Star16Pointed	Gets the Star16Pointed predefined shape.
	Star24Pointed	Gets the Star24Pointed predefined shape.
	Star32Pointed	Gets the Star32Pointed predefined shape.
	Star4Pointed	Gets the Star4Pointed predefined shape.
	Star5Pointed	Gets the Star5Pointed predefined shape.
	Star6Pointed	Gets the Star6Pointed predefined shape.
	Star7Pointed	Gets the Star7Pointed predefined shape.
	Start	Gets the Start predefined shape.
	Tape	Gets the Tape predefined shape.
	Terminator	Gets the Terminator predefined shape.
	Tetragon	Gets the Tetragon predefined shape.
	TransmittalTape	Gets the TransmittalTape predefined shape.
	Trapezoid	Gets the Trapezoid predefined shape.
	Triangle	Gets the Triangle predefined shape.

See Also

Shapes Class
MindFusion.Diagramming Namespace

Shapes Methods

Shapes.Actor Method

Gets the Actor predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Actor ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Alternative Method

Gets the Alternative predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Alternative ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow Method

Gets the Arrow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Arrow ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow1 Method

Gets the Arrow1 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Arrow1 ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow2 Method

Gets the Arrow2 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Arrow2 ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow3 Method

Gets the Arrow3 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Arrow3 ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow4 Method

Gets the Arrow4 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Arrow4 ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow5 Method

Gets the Arrow5 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Arrow5 ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow6 Method

Gets the Arrow6 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Arrow6 ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow7 Method

Gets the Arrow7 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Arrow7 ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow8 Method

Gets the Arrow8 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Arrow8 ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow9 Method

Gets the Arrow9 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Arrow9 ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BackSlash Method

Gets the BackSlash predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BackSlash ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BeginLoop Method

Gets the BeginLoop predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BeginLoop ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BowArrow Method

Gets the BowArrow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BowArrow ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnComplex Method

Gets the BpmnComplex predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnComplex ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnDataBasedXor Method

Gets the BpmnDataBasedXor predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnDataBasedXor ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndCancel Method

Gets the BpmnEndCancel predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnEndCancel ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndCompensation Method

Gets the BpmnEndCompensation predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnEndCompensation ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndError Method

Gets the BpmnEndError predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnEndError ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndLink Method

Gets the BpmnEndLink predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnEndLink ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndMessage Method

Gets the BpmnEndMessage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnEndMessage ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndMultiple Method

Gets the BpmnEndMultiple predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnEndMultiple ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndTerminate Method

Gets the BpmnEndTerminate predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnEndTerminate ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEventBasedXor Method

Gets the BpmnEventBasedXor predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnEventBasedXor ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnInclusive Method

Gets the BpmnInclusive predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnInclusive ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateCancel Method

Gets the BpmnIntermediateCancel predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnIntermediateCancel ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateCompensation Method

Gets the BpmnIntermediateCompensation predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnIntermediateCompensation ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateError Method

Gets the BpmnIntermediateError predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function BpmnIntermediateError ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateLink Method

Gets the BpmnIntermediateLink predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function BpmnIntermediateLink ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateMessage Method

Gets the BpmnIntermediateMessage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function BpmnIntermediateMessage ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateMultiple Method

Gets the BpmnIntermediateMultiple predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function BpmnIntermediateMultiple ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateRule Method

Gets the BpmnIntermediateRule predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function BpmnIntermediateRule ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateTimer Method

Gets the BpmnIntermediateTimer predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function BpmnIntermediateTimer ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnParallelFork Method

Gets the BpmnParallelFork predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnParallelFork ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnStartLink Method

Gets the BpmnStartLink predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnStartLink ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnStartMessage Method

Gets the BpmnStartMessage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnStartMessage ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnStartMultiple Method

Gets the BpmnStartMultiple predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnStartMultiple ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnStartRule Method

Gets the BpmnStartRule predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnStartRule ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnStartTimer Method

Gets the BpmnStartTimer predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function BpmnStartTimer ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Circle Method

Gets the Circle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Circle ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Cloud Method

Gets the Cloud predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Cloud ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Collate Method

Gets the Collate predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Collate ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ConeDown Method

Gets the ConeDown predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function ConeDown ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ConeUp Method

Gets the ConeUp predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function ConeUp ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ConnectedIssues Method

Gets the ConnectedIssues predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function ConnectedIssues ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.CreateRequest Method

Gets the CreateRequest predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function CreateRequest ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Cross Method

Gets the Cross predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Cross ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Cube Method

Gets the Cube predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Cube ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Cylinder Method

Gets the Cylinder predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Cylinder ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Database Method

Gets the Database predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Database ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DataTransmition Method

Gets the DataTransmition predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function DataTransmition ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DDelay Method

Gets the DDelay predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function DDelay ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Decagon Method

Gets the Decagon predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Decagon ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Decision Method

Gets the Decision predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Decision ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Decision2 Method

Gets the Decision2 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Decision2 ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DefaultFlow Method

Gets the DefaultFlow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function DefaultFlow ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Delay Method

Gets the Delay predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Delay ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DInDelay Method

Gets the DInDelay predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function DInDelay ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DirectAccessStorage Method

Gets the DirectAccessStorage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function DirectAccessStorage ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DiskStorage Method

Gets the DiskStorage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function DiskStorage ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Display Method

Gets the Display predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Display ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DividedEvent Method

Gets the DividedEvent predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function DividedEvent ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DividedProcess Method

Gets the DividedProcess predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function DividedProcess ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Document Method

Gets the Document predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Document ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Donut Method

Gets the Donut predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Donut ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DoubleArrow Method

Gets the DoubleArrow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function DoubleArrow ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DOutDelay Method

Gets the DOutDelay predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function DOutDelay ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Ellipse Method

Gets the Ellipse predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Ellipse ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.EndLoop Method

Gets the EndLoop predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function EndLoop ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ExternalOrganization Method

Gets the ExternalOrganization predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function ExternalOrganization ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ExternalProcess Method

Gets the ExternalProcess predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function ExternalProcess ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.File Method

Gets the File predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function File ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.FramedRectangle Method

Gets the FramedRectangle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function FramedRectangle ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Gate Method

Gets the Gate predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Gate ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Heart Method

Gets the Heart predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Heart ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Heptagon Method

Gets the Heptagon predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Heptagon ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Hourglass Method

Gets the Hourglass predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Hourglass ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Input Method

Gets the Input predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Input ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.InternalStorage Method

Gets the InternalStorage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function InternalStorage ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Interrupt Method

Gets the Interrupt predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Interrupt ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Interrupt2 Method

Gets the Interrupt2 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Interrupt2 ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.IsoProcess Method

Gets the IsoProcess predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function IsoProcess ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Junction Method

Gets the Junction predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Junction ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Lightning Method

Gets the Lightning predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Lightning ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.LinedDocument Method

Gets the LinedDocument predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function LinedDocument ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ManualOperation Method

Gets the ManualOperation predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function ManualOperation ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Merge Method

Gets the Merge predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Merge ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MessageFromUser Method

Gets the MessageFromUser predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function MessageFromUser ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MessageToUser Method

Gets the MessageToUser predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function MessageToUser ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Microform Method

Gets the Microform predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Microform ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MicroformProcessing Method

Gets the MicroformProcessing predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function MicroformProcessing ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MicroformRecording Method

Gets the MicroformRecording predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function MicroformRecording ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MultiDocument Method

Gets the MultiDocument predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function MultiDocument ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MultiProc Method

Gets the MultiProc predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function MultiProc ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Octagon Method

Gets the Octagon predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Octagon ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.OfflineStorage Method

Gets the OfflineStorage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function OfflineStorage ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.OffpageConnection Method

Gets the OffpageConnection predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function OffpageConnection ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.OffpageReference Method

Gets the OffpageReference predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function OffpageReference ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Or Method

Gets the Or predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Or ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Pentagon Method

Gets the Pentagon predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Pentagon ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PentagonArrow Method

Gets the PentagonArrow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function PentagonArrow ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Plaque Method

Gets the Plaque predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Plaque ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PointerArrow Method

Gets the PointerArrow predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function PointerArrow ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PrimitiveFromCall Method

Gets the PrimitiveFromCall predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function PrimitiveFromCall ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PrimitiveToCall Method

Gets the PrimitiveToCall predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function PrimitiveToCall ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Procedure Method

Gets the Procedure predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Procedure ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Process Method

Gets the Process predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Process ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Prysm Method

Gets the Prysm predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Prysm ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PunchedCard Method

Gets the PunchedCard predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function PunchedCard ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PyramidDown Method

Gets the PyramidDown predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function PyramidDown ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PyramidUp Method

Gets the PyramidUp predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function PyramidUp ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Quill Method

Gets the Quill predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Quill ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Rectangle Method

Gets the Rectangle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Rectangle ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Reversed Method

Gets the Reversed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Reversed ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.RevTriangle Method

Gets the RevTriangle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function RevTriangle ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.RevWithCirc Method

Gets the RevWithCirc predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function RevWithCirc ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.RevWithLine Method

Gets the RevWithLine predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function RevWithLine ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Rhombus Method

Gets the Rhombus predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Rhombus ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.RightTriangle Method

Gets the RightTriangle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function RightTriangle ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.RoundRect Method

Gets the RoundRect predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function RoundRect ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.RSave Method

Gets the RSave predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function RSave ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Save Method

Gets the Save predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Save ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Slash Method

Gets the Slash predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Slash ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Sort Method

Gets the Sort predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Sort ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Sphere Method

Gets the Sphere predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Sphere ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star Method

Gets the Star predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Star ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star16Pointed Method

Gets the Star16Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Star16Pointed ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star24Pointed Method

Gets the Star24Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Star24Pointed ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star32Pointed Method

Gets the Star32Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Star32Pointed ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star4Pointed Method

Gets the Star4Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Star4Pointed ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star5Pointed Method

Gets the Star5Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Star5Pointed ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star6Pointed Method

Gets the Star6Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Star6Pointed ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star7Pointed Method

Gets the Star7Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Star7Pointed ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Start Method

Gets the Start predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Start ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Tape Method

Gets the Tape predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Tape ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Terminator Method

Gets the Terminator predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Terminator ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Tetragon Method

Gets the Tetragon predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function Tetragon ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.TransmittalTape Method

Gets the TransmittalTape predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript — Copy Code

```
function TransmittalTape ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Trapezoid Method

Gets the Trapezoid predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Trapezoid ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Triangle Method

Gets the Triangle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: Shapes.js

Syntax

JavaScript Copy Code

```
function Triangle ()
```

Return Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

ShowAnchors Enumeration

Specifies when anchor points should be displayed.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
ShowAnchors = {}
```

Members

	Member name	Description
	Always	Anchor point marks are always painted.
	Never	Anchor point marks are never painted.

	Auto	Anchor point marks are shown for the node under the mouse cursor.
	Selected	Anchor point marks are painted for the selected node.

Remarks

Members of this enumeration can be assigned to the showAnchors property of Diagram.

See Also

MindFusion.Diagramming Namespace

SimplePanel Class

A layout container that arranges its child components on top of each other.

Syntax

JavaScript Copy Code

```
// class
SimplePanel.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.Container
  MindFusion.Diagramming.Panel
    MindFusion.Diagramming.SimplePanel

      MindFusion.Diagramming.Border

See Also

SimplePanel Members
MindFusion.Diagramming Namespace
Composite Nodes

SimplePanel Members

A layout container that arranges its child components on top of each other.

The following tables list the members exposed by the SimplePanel type.

Public Constructors

	Name	Description
	SimplePanel	Initializes a new instance of the SimplePanel class.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)

	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

Public Methods

	Name	Description
	getComponents	Gets the child components of this panel. (Inherited from Panel.)

See Also

SimplePanel Class
MindFusion.Diagramming Namespace

SimplePanel Constructor

Initializes a new instance of the SimplePanel class.

Namespace: MindFusion.Diagramming
Assembly: SimplePanel.js

Syntax

JavaScript Copy Code

```
function SimplePanel ()
```

See Also

SimplePanel Class
MindFusion.Diagramming Namespace

SimpleShape Enumeration

Defines the possible values for shape property of tables and containers.

Namespace: MindFusion.Diagramming
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
SimpleShape = {}
```

Members

	Member name	Description
	Rectangle	Sets node's shape to rectangle.
	RoundedRectangle	Sets node's shape to rounded rectangle.

See Also

MindFusion.Diagramming Namespace

SinglePointerController Class

Processes mouse or single-touch events targeting a diagram element.

Syntax

JavaScript — Copy Code

```
// class
SinglePointerController.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.SinglePointerController
- MindFusion.Diagramming.CreateLinkController
- MindFusion.Diagramming.CreateNodeController
- MindFusion.Diagramming.CreateSelectionController
- MindFusion.Diagramming.ModifyLinkController
- MindFusion.Diagramming.ModifyNodeController
- MindFusion.Diagramming.ModifySelectionController
- MindFusion.Diagramming.PanController

See Also

SinglePointerController Members
MindFusion.Diagramming Namespace

SinglePointerController Members

Processes mouse or single-touch events targeting a diagram element.

The following tables list the members exposed by the SinglePointerController type.

Public Constructors

	Name	Description
	SinglePointerController	Initializes a new instance of the SinglePointerController class.

Public Methods

	Name	Description
	cancel	Called when the user stops drawing or modifying a diagram element, rejecting the operation.
	commit	Called when the user stops drawing or modifying a diagram element, accepting the operation.
	drawInteraction	Called to draw current state of user interaction.
	move	Called continuously while the user draws with the mouse or touch.
	start	Called when the user starts drawing or modifying a diagram element.
	validate	Called to validate the operation at current pointer position.

See Also

SinglePointerController Class
MindFusion.Diagramming Namespace

SinglePointerController Constructor

Initializes a new instance of the SinglePointerController class.

Namespace: MindFusion.Diagramming
Assembly: SinglePointerController.js

Syntax

JavaScript Copy Code

```
function SinglePointerController (modifiedItem, adjustmentHandle)
```

Parameters

modifiedItem
DiagramItem. The item that will be modified by this controller.

adjustmentHandle
Object. An object representing the adjustment handle used to modify the item.

See Also

SinglePointerController Class
MindFusion.Diagramming Namespace

SinglePointerController Methods

SinglePointerController.cancel Method

Called when the user stops drawing or modifying a diagram element, rejecting the operation.

Namespace: MindFusion.Diagramming
Assembly: SinglePointerController.js

Syntax

JavaScript Copy Code

```
function cancel (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

SinglePointerController Members
SinglePointerController Class
MindFusion.Diagramming Namespace

SinglePointerController.commit Method

Called when the user stops drawing or modifying a diagram element, accepting the operation.

Namespace: MindFusion.Diagramming
Assembly: SinglePointerController.js

Syntax

JavaScript Copy Code

```
function commit (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

SinglePointerController Members
SinglePointerController Class
MindFusion.Diagramming Namespace

SinglePointerController.drawInteraction Method

Called to draw current state of user interaction.

Namespace: MindFusion.Diagramming
Assembly: SinglePointerController.js

Syntax

JavaScript — Copy Code

```
function drawInteraction (context)
```

Parameters

context
DrawingContext2D. The drawing context for diagram's underlying canvas.

See Also

SinglePointerController Members
SinglePointerController Class
MindFusion.Diagramming Namespace

SinglePointerController.move Method

Called continuously while the user draws with the mouse or touch.

Namespace: MindFusion.Diagramming
Assembly: SinglePointerController.js

Syntax

JavaScript — Copy Code

```
function move (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

SinglePointerController Members
SinglePointerController Class
MindFusion.Diagramming Namespace

SinglePointerController.start Method

Called when the user starts drawing or modifying a diagram element.

Namespace: MindFusion.Diagramming
Assembly: SinglePointerController.js

Syntax

JavaScript — Copy Code

```
function start (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

See Also

SinglePointerController Members
SinglePointerController Class
MindFusion.Diagramming Namespace

SinglePointerController.validate Method

Called to validate the operation at current pointer position.

Namespace: MindFusion.Diagramming
Assembly: SinglePointerController.js

Syntax

JavaScript Copy Code

```
function validate (position)
```

Parameters

position
Point. A point specified in diagram coordinates.

Return Value

Boolean. true to accept the operation, or false otherwise.

See Also

SinglePointerController Members
SinglePointerController Class
MindFusion.Diagramming Namespace

StackPanel Class

A layout container that arranges its child components stacked horizontally or vertically.

Syntax

JavaScript Copy Code

```
// class
StackPanel.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.Container
  MindFusion.Diagramming.Panel
    MindFusion.Diagramming.StackPanel

See Also

StackPanel Members
MindFusion.Diagramming Namespace
Composite Nodes

StackPanel Members

A layout container that arranges its child components stacked horizontally or vertically.

The following tables list the members exposed by the StackPanel type.

Public Constructors

	Name	Description
	StackPanel	Initializes a new instance of the **StackPanel** class.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	orientation	Gets or sets the stack orientation.
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

Public Methods

	Name	Description
	getComponents	Gets the child components of this panel. (Inherited from Panel.)
	getOrientation	Gets the stack orientation.
	setOrientation	Sets the stack orientation.

See Also

StackPanel Class
MindFusion.Diagramming Namespace

StackPanel Constructor

Initializes a new instance of the StackPanel class.

Namespace: MindFusion.Diagramming
Assembly: StackPanel.js

Syntax

JavaScript Copy Code

```
function StackPanel ()
```

See Also

StackPanel Class
MindFusion.Diagramming Namespace

StackPanel Methods

StackPanel.getOrientation Method

Gets the stack orientation.

Namespace: MindFusion.Diagramming
Assembly: StackPanel.js

Syntax

JavaScript — Copy Code

```
function getOrientation ()
```

Return Value

A member of the Orientation enumeration.

See Also

StackPanel Members
StackPanel Class
MindFusion.Diagramming Namespace

StackPanel.setOrientation Method

Sets the stack orientation.

Namespace: MindFusion.Diagramming
Assembly: StackPanel.js

Syntax

JavaScript — Copy Code

```
function setOrientation (value)
```

Parameters

value
A member of the Orientation enumeration.

See Also

StackPanel Members
StackPanel Class
MindFusion.Diagramming Namespace

StackPanel Properties

StackPanel.orientation Property

Gets or sets the stack orientation.

Namespace: MindFusion.Diagramming
Assembly: StackPanel.js

Syntax

JavaScript — Copy Code

```
get orientation() {}
set orientation(value) {}
```

Property Value

Orientation

A member of the Orientation enumeration.

See Also

StackPanel Members
StackPanel Class
MindFusion.Diagramming Namespace

Style Class

Represents a set of appearance properties whose values can be inherited from parent objects if not set locally for an item.

Syntax

JavaScript Copy Code

```
// class
Style.prototype = {}
```

Remarks

An instance of this class can be used to set items' appearance properties as a single unit via their style property. In order for style attributes to be used, the respective local appearance properties of a DiagramItem objects must have undefined or **null** values. If an item's **style** does not define a value for some attribute, it is looked up in the styling hierarchy, first in the diagram's style, next in the current theme, and finally in the set of default styles defined for each item class.

Inheritance Hierarchy

MindFusion.Diagramming.Style

See Also

Style Members
MindFusion.Diagramming Namespace

Style Members

Represents a set of appearance properties whose values can be inherited from parent objects if not set locally for an item.

The following tables list the members exposed by the Style type.

Public Constructors

	Name	Description
	Style	Initializes a new instance of the Style class.

Public Properties

	Name	Description
	backBrush	Gets or sets the brush used to paint diagram background.
	brush	Gets or sets the brush used to fill the interior of diagram items.
	captionBackBrush	Gets or sets the brush used to fill nodes' caption bars.
	fontName	Gets or sets the font name used to draw text of diagram items.
	fontSize	Gets or sets the font size used to draw text of diagram items.
	fontStyle	Gets or sets the font style used to draw text of diagram items.
	nodeEffects	Gets or sets a list of effects applied to diagram nodes.
	selectedBrush	Gets or sets the brush used to paint the background of selected items.

	selectedStroke	Gets or sets the color used to stroke outlines of selected items.
	shadowColor	Gets or sets the color used to draw shadows of diagram items.
	stroke	Gets or sets the color used to stroke outlines of diagram items.
	strokeDashStyle	Gets or sets the line dash pattern applied when stroking outlines of diagram items.
	strokeThickness	Gets or sets the line width applied when stroking outlines of diagram items.
	textColor	Gets or sets the color used to draw text of diagram items.

See Also

Style Class
MindFusion.Diagramming Namespace

Style Constructor

Initializes a new instance of the Style class.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
function Style ([prototype])
```

Parameters

prototype
Optional.

See Also

Style Class
MindFusion.Diagramming Namespace

Style Properties

Style.backBrush Property

Gets or sets the brush used to paint diagram background.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get backBrush() {}
set backBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.brush Property

Gets or sets the brush used to fill the interior of diagram items.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get brush() {}
set brush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.captionBackBrush Property

Gets or sets the brush used to fill nodes' caption bars.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get captionBackBrush() {}
set captionBackBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

Remarks

This value is applied to TableNode and ContainerNode objects.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.fontName Property

Gets or sets the font name used to draw text of diagram items.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get fontName() {}
set fontName(value) {}
```

Property Value

String

A string containing font name.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.fontSize Property

Gets or sets the font size used to draw text of diagram items.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get fontSize() {}
set fontSize(value) {}
```

Property Value

Number

A number specifying font size.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.fontStyle Property

Gets or sets the font style used to draw text of diagram items.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get fontStyle() {}
set fontStyle(value) {}
```

Property Value

FontStyle

A combination of FontStyle enumeration members.

See Also

Style.nodeEffects Property

Gets or sets a list of effects applied to diagram nodes.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get nodeEffects() {}
set nodeEffects(value) {}
```

Property Value

Array

An array of NodeEffect objects.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.selectedBrush Property

Gets or sets the brush used to paint the background of selected items.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get selectedBrush() {}
set selectedBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.selectedStroke Property

Gets or sets the color used to stroke outlines of selected items.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get selectedStroke() {}
set selectedStroke(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.shadowColor Property

Gets or sets the color used to draw shadows of diagram items.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get shadowColor() {}
set shadowColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.stroke Property

Gets or sets the color used to stroke outlines of diagram items.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get stroke() {}
set stroke(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.strokeDashStyle Property

Gets or sets the line dash pattern applied when stroking outlines of diagram items.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get strokeDashStyle() {}
set strokeDashStyle(value) {}
```

Property Value

DashStyle

A member of the DashStyle enumeration.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.strokeThickness Property

Gets or sets the line width applied when stroking outlines of diagram items.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get strokeThickness() {}
set strokeThickness(value) {}
```

Property Value

Number

A number specifying stroke thickness.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.textColor Property

Gets or sets the color used to draw text of diagram items.

Namespace: MindFusion.Diagramming
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

SvgContent Class

SvgContent represents the markup code of an SVG drawing.

Syntax

JavaScript Copy Code

```
// class
SvgContent.prototype = {}
```

Remarks

SvgContent objects are assigned to the content property of SvgNode.

Inheritance Hierarchy

MindFusion.Diagramming.SvgContent

See Also

SvgContent Members
MindFusion.Diagramming Namespace

SvgContent Members

SvgContent represents the markup code of an SVG drawing.

The following tables list the members exposed by the **SvgContent** type.

Public Methods

	Name	Description
	parse	Loads SVG code from the specified file.
	parseSvg	Loads SVG code from the specified string.

See Also

SvgContent Class
MindFusion.Diagramming Namespace

SvgContent Methods

SvgContent.parse Method

Loads SVG code from the specified file.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript Copy Code

```
function parse (value)
```

Parameters

value
String. A string specifying the SVG file location.

Remarks

The MIME type of .svg files on server should be set to image/svg+xml.

See Also

SvgContent Members
SvgContent Class
MindFusion.Diagramming Namespace

SvgContent.parseSvg Method

Loads SVG code from the specified string.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript Copy Code

```
function parseSvg (value)
```

Parameters

value
String. A string containing the SVG markup code.

See Also

SvgContent Members
SvgContent Class
MindFusion.Diagramming Namespace

SvgExporter Class

Creates SVG (Scalable Vector Graphics) drawings from the content of Diagram objects.

Syntax

JavaScript Copy Code

```
// class
SvgExporter.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.SvgExporter

See Also

SvgExporter Members
MindFusion.Diagramming Namespace

SvgExporter Members

Creates SVG (Scalable Vector Graphics) drawings from the content of Diagram objects.

The following tables list the members exposed by the SvgExporter type.

Public Constructors

	Name	Description
	SvgExporter	Initializes a new instance of the **SvgExporter** class.

Public Methods

	Name	Description
	exportElement	Overloaded.
	exportString	Creates an SVG document representing the specified diagram.

See Also

SvgExporter Class
MindFusion.Diagramming Namespace

SvgExporter Constructor

Initializes a new instance of the SvgExporter class.

Namespace: MindFusion.Diagramming
Assembly: SvgExporter.js

Syntax

JavaScript Copy Code

```
function SvgExporter ()
```

See Also

SvgExporter Class
MindFusion.Diagramming Namespace

SvgExporter Methods

SvgExporter.exportElement Method

Overload List

Name	Description
SvgExporter.exportElement (Diagram)	Creates an SVG element representing the specified diagram.
SvgExporter.exportElement (Diagram, Document)	Creates an SVG element representing the specified diagram.

See Also

SvgExporter Members
SvgExporter Class
MindFusion.Diagramming Namespace

SvgExporter.exportElement Method (Diagram)

Creates an SVG element representing the specified diagram.

Namespace: MindFusion.Diagramming
Assembly: SvgExporter.js

Syntax

JavaScript Copy Code

```
function exportElement (diagram)
```

Parameters

diagram
A reference to the diagram to be exported.

See Also

exportElement Method Overload List
SvgExporter Members
SvgExporter Class
MindFusion.Diagramming Namespace

SvgExporter.exportElement Method (Diagram, Document)

Creates an SVG element representing the specified diagram.

Namespace: MindFusion.Diagramming
Assembly: SvgExporter.js

Syntax

JavaScript Copy Code

```
function exportElement (diagram, [parentDoc])
```

Parameters

diagram
Diagram. A reference to the diagram to be exported.

parentDoc
Optional.
Document. The document to export to.

Return Value

Object. A DOM SVG element.

See Also

exportElement Method Overload List
SvgExporter Members
SvgExporter Class
MindFusion.Diagramming Namespace

SvgExporter.exportString Method

Creates an SVG document representing the specified diagram.

Namespace: MindFusion.Diagramming
Assembly: SvgExporter.js

Syntax

JavaScript Copy Code

```
function exportString (diagram)
```

Parameters

diagram
Diagram. A reference to the diagram to be exported.

Return Value

Object. A string containing the SVG document's content.

See Also

SvgExporter Members
SvgExporter Class
MindFusion.Diagramming Namespace

SvgNode Class

SvgNode instances are diagram nodes that can render SVG drawings.

Syntax

JavaScript Copy Code

```
// class
SvgNode.prototype = {}
```

Remarks

An **SvgNode** instance represents nodes that can display SVG drawings in addition to the rendering provided by the base ShapeNode class. The SVG drawing of a node is specified through its content property. The SVG graphics are loaded and associated with the node through an SvgContent object, which should be initialized and then assigned to **content**. Note that the MIME type of .svg files on server should be set to image/svg+xml.

Example

The following code illustrates how to load SVG graphics from a server file and associate it with a new **SvgNode**:

JavaScript Copy Code

```
var node = diagram.factory.createSvgNode(bounds);
node.transparent = true; // hide ShapeNode geometry

var svg = new SvgContent();
svg.parse("image.svg");
node.content = svg;
```

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.ShapeNode
      MindFusion.Diagramming.SvgNode

See Also

SvgNode Members
MindFusion.Diagramming Namespace

SvgNode Members

SvgNode instances are diagram nodes that can render SVG drawings.

The following tables list the members exposed by the **SvgNode** type.

Public Constructors

	Name	Description
	SvgNode	Initializes a new instance of the SvgNode class with default values supplied from the specified Diagram.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowFlip	Gets or sets a value indicating whether the node's shape flips when an edge is dragged over its opposite edge. (Inherited from ShapeNode.)
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	content	Gets or sets an SvgContent instance representing the SVG drawing rendered in this node.
	contentAlign	Gets or sets the alignment of the **SvgContent** displayed in this node.
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	CustomDraw	Gets or sets the type of custom drawing performed on this shape node. (Inherited from ShapeNode.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled. (Inherited from ShapeNode.)

	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	FlipX	Gets or sets a value indicating whether the node's shape is flipped horizontally. (Inherited from ShapeNode.)
	FlipY	Gets or sets a value indicating whether the node's shape is flipped vertically. (Inherited from ShapeNode.)
	Font	Gets or sets the font used to render the node's text. (Inherited from ShapeNode.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	Image	Gets or sets the image that is displayed in this shape node. (Inherited from ShapeNode.)
	ImageAlign	Gets or sets the alignment of the image displayed in this shape node. (Inherited from ShapeNode.)
	ImagePadding	Gets or sets the padding distance between node's borders and **Image**. (Inherited from ShapeNode.)
	ImageUrl	Gets or sets a URL used to load an image into this node. (Inherited from ShapeNode.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)

	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	PlainText	Gets the plain text of this shape node. (Inherited from ShapeNode.)
	PolygonalTextLayout	Gets or sets a value indicating whether the node's Text is formatted inside non-rectangular outline. (Inherited from ShapeNode.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotateImage	Gets or sets a value indicating whether the **Image** should be rotated when the shape node is rotated. (Inherited from ShapeNode.)
	RotateText	Gets or sets a value indicating whether the **Text** should be rotated when the shape node is rotated. (Inherited from ShapeNode.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Shape	Gets or sets a reference to the node's geometric shape definition. (Inherited from ShapeNode.)
	ShapeOrientation	Gets or sets the angle specifying the orientation with which a shape template is applied to the node bounds. (Inherited from ShapeNode.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed inside the shape node. (Inherited from ShapeNode.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of the text that is displayed inside shape nodes. (Inherited from ShapeNode.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from ShapeNode.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Transparent	Gets or sets a value indicating whether this shape node is transparent. (Inherited from ShapeNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)

	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	clone	ShapeNode.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from ShapeNode.)
	CreateConnectionPoint	DiagramNode.CreateConnectionPoint override. (Inherited from ShapeNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramNode.DrawHandles override. Renders the adjustment handles of this item. (Inherited from ShapeNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Draws the node on the specified IGraphics surface. (Inherited from ShapeNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified **IGraphics** surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Draws the shadow of this node on the specified **IGraphics** surface. (Inherited from ShapeNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	fromJson	ShapeNode.fromJson override.

	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from ShapeNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	Node.GetOutlinePoly override. (Inherited from ShapeNode.)
	GetRepaintRect	DiagramNode.GetRepaintRect override. (Inherited from ShapeNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramNode.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from ShapeNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	init	Returns a Builder object used to configure this SvgNode instance.
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	loadFromXml	DiagramNode.LoadFromXml override. Loads the node's content from an XML element.
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)

	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	DiagramItem.OnLoad override. (Inherited from ShapeNode.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	ResizeToFitImage	Makes the shape node as big as the image it displays. (Inherited from ShapeNode.)
	ResizeToFitText	Makes the shape node big enough to display its text without clipping. (Inherited from ShapeNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveToXml	Saves the node's content into an XML element.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	toJson	ShapeNode.toJson override.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	With	Returns a Builder object used to configure and create new **SvgNode** instances.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)

	AllowModify	DiagramNode.AllowModify override. Invoked during interactive modification of items to test whether the operation can be completed in the current context. (Inherited from ShapeNode.)
	CancelModify	DiagramNode.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from ShapeNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramNode.CompleteModify override. Invoked when the item modification has been completed. (Inherited from ShapeNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from ShapeNode.)
	CreateState	DiagramNode.CreateState override; creates a ShapeNodeState instance. (Inherited from ShapeNode.)
	Dispose	Disposes any disposable resources managed by this shape. (Inherited from ShapeNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetImageRect	Overloaded. Gets the destination rectangle of the node's image. (Inherited from ShapeNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a GraphicsPath object for this node as though it is contained in the specified rectangle. (Inherited from ShapeNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object. (Inherited from ShapeNode.)
	LoadFromXml	Loads the item content from an XML element. (Inherited from ShapeNode.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	Called when the item is added to a **Diagram**, either by users, programmatically or via undo / redo. (Inherited from ShapeNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	Called when the Diagram's MeasureUnit changes. (Inherited from ShapeNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)

	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	DiagramNode.OnRotationAngleChanged override. (Inherited from ShapeNode.)
	OnStyleChanged	DiagramItem.OnStyleChanged override. (Inherited from ShapeNode.)
	OnTextChanged	DiagramItem.OnTextChanged override. (Inherited from ShapeNode.)
	OnTextFormatChanged	DiagramItem.OnTextFormatChanged override. (Inherited from ShapeNode.)
	OnUpdateBounds	DiagramNode.OnUpdateBounds override. (Inherited from ShapeNode.)
	OnZoomChanged	Called when the Diagram's **ZoomFactor** changes. (Inherited from ShapeNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramNode.RestoreProperties override. (Inherited from ShapeNode.)
	RestoreState	DiagramNode.RestoreState override. Restores the state of this diagram item. (Inherited from ShapeNode.)
	SaveProperties	DiagramNode.SaveProperties override. (Inherited from ShapeNode.)
	SaveState	DiagramNode.SaveState override. Saves the state of this diagram item. (Inherited from ShapeNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from ShapeNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from ShapeNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramNode.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from ShapeNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)

	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	**DiagramNode.UpdateModify** override. Invoked during interactive modification of this item. (Inherited from ShapeNode.)

See Also

SvgNode Class
MindFusion.Diagramming Namespace

SvgNode Constructor

Initializes a new instance of the SvgNode class with default values supplied from the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript Copy Code

```
function SvgNode ([parent])
```

Parameters

parent
Optional.
Diagram. The Diagram from which to obtain default values for the node properties.

See Also

SvgNode Class
MindFusion.Diagramming Namespace

SvgNode Methods

SvgNode.clone Method

ShapeNode.clone override.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript Copy Code

```
function clone ()
```

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.fromJson Method

ShapeNode.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript — Copy Code

```
function fromJson ()
```

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.init Method

Returns a Builder object used to configure this SvgNode instance.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript — Copy Code

```
function init ()
```

Return Value

A Builder instance.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.loadFromXml Method

DiagramNode.LoadFromXml override. Loads the node's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript — Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

SvgNode Members
SvgNode Class

SvgNode.saveToXml Method

Saves the node's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.toJson Method

ShapeNode.toJson override.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.With Method

Returns a Builder object used to configure and create new SvgNode instances.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript Copy Code

```
function With (diagram)
```

Parameters

diagram

Return Value

A Builder instance.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode Properties

SvgNode.content Property

Gets or sets an SvgContent instance representing the SVG drawing rendered in this node.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript Copy Code

```
get content() {}
set content(value) {}
```

Property Value

SvgContent

An instance of the SvgContent class.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace
contentAlign Property

SvgNode.contentAlign Property

Gets or sets the alignment of the SvgContent displayed in this node.

Namespace: MindFusion.Diagramming
Assembly: SvgNode.js

Syntax

JavaScript Copy Code

```
get contentAlign() {}
set contentAlign(value) {}
```

Property Value

ImageAlign

A member of the ImageAlign enumeration.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

TableColumn Class

Represents a column of a TableNode.

Syntax

JavaScript Copy Code

```
// class
TableColumn.prototype = {}
```

Remarks

Call the getColumn method of TableNode to get the **TableColumn** instance at specified position. In order to add or remove columns, set table's columnCount property, or call methods such as addColumn, insertColumn, deleteColumn, redimTable.

Inheritance Hierarchy

MindFusion.Diagramming.TableColumn

See Also

TableColumn Members
MindFusion.Diagramming Namespace
TableRow Class

TableColumn Members

Represents a column of a TableNode.

The following tables list the members exposed by the TableColumn type.

Public Constructors

	Name	Description
	TableColumn	Initializes a new instance of the TableColumn class.

Public Properties

	Name	Description
	columnStyle	Gets or sets the column style.
	width	Gets or sets the width of this column.

See Also

TableColumn Class
MindFusion.Diagramming Namespace

TableColumn Constructor

Initializes a new instance of the TableColumn class.

Namespace: MindFusion.Diagramming
Assembly: TableColumn.js

Syntax

JavaScript Copy Code

```
function TableColumn ()
```

See Also

TableColumn Class
MindFusion.Diagramming Namespace

TableColumn Properties

TableColumn.columnStyle Property

Gets or sets the column style.

Namespace: MindFusion.Diagramming
Assembly: TableColumn.js

Syntax

JavaScript Copy Code

```
get columnStyle() {}
set columnStyle(value) {}
```

Property Value

ColumnStyle

A member of the ColumnStyle enumeration.

See Also

TableColumn Members
TableColumn Class
MindFusion.Diagramming Namespace

TableColumn.width Property

Gets or sets the width of this column.

Namespace: MindFusion.Diagramming
Assembly: TableColumn.js

Syntax

JavaScript Copy Code

```
get width() {}
set width(value) {}
```

Property Value

Number

A number specifying the width.

See Also

TableColumn Members
TableColumn Class
MindFusion.Diagramming Namespace

TableConnectionPoint Class

A class that represents a connection between an edge and a node in a graph.

Syntax

JavaScript Copy Code

```
// class
TableConnectionPoint.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.ConnectionPoint
MindFusion.Diagramming.TableConnectionPoint

See Also

TableConnectionPoint Members
MindFusion.Diagramming Namespace

TableConnectionPoint Members

A class that represents a connection between an edge and a node in a graph.

The following tables list the members exposed by the TableConnectionPoint type.

Public Properties

	Name	Description
	endPoint	Calculates the coordinates of this connection point expressed in document coordinates.
	RelativePosition	Gets or sets the position of this connection point, relative to the associated node's bounding rectangle, expressed as percentage. (Inherited from ConnectionPoint.)
	Row	Gets the index of a constituent item within the associated node where the link represented by this connection point is connected, or -1, if the link is connected to the node itself. (Inherited from ConnectionPoint.)

Protected Properties

	Name	Description
	Incoming	Gets a flag indicating whether this connection point is at the origin or the destination of the associated link. (Inherited from ConnectionPoint.)
	Link	Gets the link associated with this connection point. (Inherited from ConnectionPoint.)
	Node	Gets the node associated with this connection point. (Inherited from ConnectionPoint.)

Public Methods

	Name	Description
	addLinkToNode	Adds the link associated with this connection point to the associated node.
	AddLinkToNode	Adds the link associated with this connection point to the associated node. (Inherited from ConnectionPoint.)
	CalculateIntersections	Determines whether the end of a link should be aligned to the point where the node contour is intersected by the line connecting the centers of the connected nodes. (Inherited from ConnectionPoint.)
	CanMoveLink	Checks whether the specified point, expressed in document coordinates, is a valid location for this connection point. (Inherited from ConnectionPoint.)
	GetAnchorPos	Overloaded. (Inherited from ConnectionPoint.)

	GetEndPoint	Calculates the coordinates of the this connection point expressed in document coordinates. (Inherited from ConnectionPoint.)
	getInitialPoint	Returns a point (usually the center of the bounding rectangle) contained within the associated node.
	GetInitialPoint	Returns a point (usually the center of the bounding rectangle) contained within the associated node. (Inherited from ConnectionPoint.)
	getIntersection	Calculates the intersection point between the specified line segment and the contour of the node associated with this connection point.
	GetIntersection	Calculates the intersection point between the specified line segment and the contour of the node associated with this connection point. (Inherited from ConnectionPoint.)
	getNearestBorderPoint	Returns the nearest point lying on the node border.
	GetNodeRect	Returns the bounding rectangle of the node associated with this connection point. (Inherited from ConnectionPoint.)
	linkChanges	Checks whether the link would change if attached to the specified node at the specified point.
	LinkChanges	Checks whether the relationship represented by the associated link would change if the link is attached to the specified node at the specified point. (Inherited from ConnectionPoint.)
	NearestAnchorPoint	Returns the anchor point closest to the specified point from the node's associated anchor pattern. (Inherited from ConnectionPoint.)
	NodesIntersect	Checks whether the node associated with the current connection point and the node associated with the specified connection point intersect. (Inherited from ConnectionPoint.)
	removeLinkFromNode	Removes the link associated with this connection point from the associated node.
	RemoveLinkFromNode	Removes the link associated with this connection point from the associated node. (Inherited from ConnectionPoint.)
	SameNode	Checks whether the current connection point and the specified connection point are associated with the same node. (Inherited from ConnectionPoint.)
	saveEndRelative	Updates the coordinates of the connection point represented by this link, relative to the bounding rectangle of the associated node.
	SaveEndRelative	Updates the coordinates of the connection point represented by this link, relative to the bounding rectangle of the associated node. (Inherited from ConnectionPoint.)

Protected Methods

	Name	Description
	GetNearestBorderPoint	Returns the nearest point lying on the node border. (Inherited from ConnectionPoint.)
	UpdateFromPoint	Updates the connection state according to the specified point, without changing the related node. (Inherited from ConnectionPoint.)

See Also

TableConnectionPoint Class
MindFusion.Diagramming Namespace

TableConnectionPoint Methods

TableConnectionPoint.addLinkToNode Method

Adds the link associated with this connection point to the associated node.

Namespace: MindFusion.Diagramming
Assembly: TableConnectionPoint.js

Syntax

JavaScript Copy Code

```
function addLinkToNode ()
```

See Also

TableConnectionPoint Members
TableConnectionPoint Class
MindFusion.Diagramming Namespace

TableConnectionPoint.getInitialPoint Method

Returns a point (usually the center of the bounding rectangle) contained within the associated node.

Namespace: MindFusion.Diagramming
Assembly: TableConnectionPoint.js

Syntax

JavaScript Copy Code

```
function getInitialPoint ()
```

Return Value

Point. A Point contained within the node.

See Also

TableConnectionPoint Members
TableConnectionPoint Class
MindFusion.Diagramming Namespace

TableConnectionPoint.getIntersection Method

Calculates the intersection point between the specified line segment and the contour of the node associated with this connection point.

Namespace: MindFusion.Diagramming
Assembly: TableConnectionPoint.js

Syntax

JavaScript Copy Code

```
function getIntersection (point1, point2)
```

Parameters

point1
Point. The first point of the segment.

point2
Point. The second point of the segment.

Return Value

Point. A Point object representing the intersection point.

See Also

TableConnectionPoint Members
TableConnectionPoint Class
MindFusion.Diagramming Namespace

TableConnectionPoint.getNearestBorderPoint Method

Returns the nearest point lying on the node border.

Namespace: MindFusion.Diagramming
Assembly: TableConnectionPoint.js

Syntax

JavaScript Copy Code

```
function getNearestBorderPoint (point)
```

Parameters

point
Point. A Point instance specifying location in the diagram.

Return Value

AnchorPointDetails. An AnchorPointDetails specifying the anchor point properties.

See Also

TableConnectionPoint Members
TableConnectionPoint Class
MindFusion.Diagramming Namespace

TableConnectionPoint.linkChanges Method

Checks whether the link would change if attached to the specified node at the specified point.

Namespace: MindFusion.Diagramming
Assembly: TableConnectionPoint.js

Syntax

JavaScript Copy Code

```
function linkChanges (item, point)
```

Parameters

item
DiagramItem. A DiagramItem under the mouse cursor.

point
Point. A point where the link will be connected.

Return Value

Boolean. true if the link would connect to a different entity if dropped at the specified location; otherwise, false.

See Also

TableConnectionPoint Members
TableConnectionPoint Class
MindFusion.Diagramming Namespace

TableConnectionPoint.removeLinkFromNode Method

Removes the link associated with this connection point from the associated node.

Namespace: MindFusion.Diagramming
Assembly: TableConnectionPoint.js

Syntax

JavaScript Copy Code

```
function removeLinkFromNode ()
```

See Also

TableConnectionPoint Members
TableConnectionPoint Class
MindFusion.Diagramming Namespace

TableConnectionPoint.saveEndRelative Method

Updates the coordinates of the connection point represented by this link, relative to the bounding rectangle of the associated node.

Namespace: MindFusion.Diagramming
Assembly: TableConnectionPoint.js

Syntax

JavaScript Copy Code

```
function saveEndRelative ()
```

See Also

TableConnectionPoint Members
TableConnectionPoint Class
MindFusion.Diagramming Namespace

TableConnectionPoint Properties

TableConnectionPoint.endPoint Property

Calculates the coordinates of this connection point expressed in document coordinates.

Namespace: MindFusion.Diagramming
Assembly: TableConnectionPoint.js

Syntax

JavaScript Copy Code

```
get endPoint() {}
set endPoint(value) {}
```

Property Value

Point

A Point that represents the coordinate.

See Also

TableConnectionPoint Members
TableConnectionPoint Class
MindFusion.Diagramming Namespace

TableNode Class

TableNode instances are diagram nodes that can be used to display tabular or list data.

Syntax

JavaScript Copy Code

```
// class
TableNode.prototype = {}
```

Remarks

Tables consist of Cell instances distributed across rows and columns. The number of rows and columns can be set by calling the redimTable method. Cells can be made to span several rows and columns by setting their rowSpan and columnSpan properties to values larger than one.

Rows can be assigned to distinct sections of the table. Each section can be collapsed or expanded, hiding or showing all rows in the section except the header one. To define a section, set a header row. A section consists of all rows after a header row and spans to the next header. Each header row displays a [±] button that allows expanding or collapsing the section interactively. The button appears on the left of the first cell of a header row. Clicking that button raises the tableSectionCollapsed and tableSectionExpanded events. Sections can be collapsed or expanded programmatically by setting the expanded property of their header rows.

The structure of a table can be dynamically changed by adding and deleting rows and columns. This can be done with the addColumn and addRow methods as well as with the insertColumn, insertRow and **redimTable** methods. deleteRow and deleteColumn methods delete existing rows and columns.

Table cells can display text as specified via their text, textColor and textAlignment properties. Cells can also contain images, as set via the image and imageAlign properties.

Table nodes can show a caption in their title bar, whose text is set via the text property inherited from DiagramItem. The height of the caption bar is set via the captionHeight property.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.TableNode

See Also

TableNode Members
MindFusion.Diagramming Namespace

TableNode Members

TableNode instances are diagram nodes that can be used to display tabular or list data.

The following tables list the members exposed by the **TableNode** type.

Public Constructors

	Name	Description
	TableNode	Initializes a new instance of the TableNode class with default values supplied from the specified Diagram.

Public Fields

	Name	Description
	useScrollBars	Set this flag to show full scrollbars for scrollable tables instead of up/down buttons in captions.

Protected Fields

	Name	Description

	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	allowResizeColumns	Gets or sets a value indicating whether users are allowed to resize table columns.
	allowResizeRows	Gets or sets a value indicating whether users are allowed to resize table rows.
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	captionBackBrush	Gets or sets an object specifying how to fill the caption bar.
	captionFont	Gets or sets the font used to render the node's caption text.
	captionHeight	Gets or sets the height of the node's caption area.
	cellFrameStyle	Gets or sets the style of cells' frame lines.
	cellStyle	Gets or sets the style used to draw table's cells.
	cellTextColor	Gets or sets the color used to fill cells text.
	columnCount	Gets or sets the number of columns in the table.
	connectionStyle	Gets or sets whether drawing a link from/to this node should connectan individual table row, or the whole table as an integral entity.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	currentScrollRow	Gets or sets the table's current scroll position.
	dividerStroke	Gets or sets the color used to stroke the divider line between node's caption bar and content area. (Inherited from CaptionedNode.)
	dividerStrokeDashArray	Gets or sets an array of values specifying the dash pattern of the node's divider line. (Inherited from CaptionedNode.)
	dividerStrokeDashOffset	Gets or sets the the distance between the line start and the beginning of the dash pattern. (Inherited from CaptionedNode.)
	dividerStrokeDashStyle	Gets or sets the line dash pattern applied when stroking node's divider line. (Inherited from CaptionedNode.)
	dividerStrokeThickness	Gets or sets the width of the divider line between node's caption bar and content area. (Inherited from CaptionedNode.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)

	effectiveDividerStrokeThickness	Gets the effective line thickness that should be applied when stroking this node's divider line. (Inherited from CaptionedNode.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	enableStyledText	Gets or sets a value indicating whether styled text rendering is enabled.
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)

	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	rowAnchorPattern	Gets or sets the default anchor pattern for the rows of a table.
	rowCount	Gets or sets the number of rows in the table.
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	scrollable	Gets or sets whether the user is allowed to scroll the table rows.
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	shape	Gets or sets the shape of the node's outline.
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	addColumn	Adds a new column to the table.
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	addRow	Adds a new row to the table.
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	cellFromPoint	Returns the cell that contains the specified point.
	clone	DiagramNode.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	containsPoint	Checks if the **TableNode** contains the specified point.
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	createConnectionPoint	Creates a connection point between this node and the specified link at the specified point.
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	deleteColumn	Deletes the specified column.
	deleteRow	Deletes the specified row.
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)

	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	fromJson	DiagramNode.fromJson override.
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	getCell	Returns a reference to the cell located at the specified column and row of this table.
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	getColumn	Returns a reference to the column at the specified position within the table's list of columns.
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	getOutline	Gets an array of points that define the outline of this node shape.
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	getParentRow	Returns the number of the first header row before thespecified row. If the specified row is a header rowitself, returns 'row'. If there is no header rowbefore this one, the return value is -1.
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	getRow	Returns a reference to the row at the specified position within the table's list of rows.
	getRowRect	Returns the bounding rectangle of the specified row.
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	insertColumn	Inserts a new column at the specified position.

	insertRow	Inserts a new row at the specified position.
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	isRowCollapsed	Checks whether the specified row is collapsed.Collapsed are all rows whose header row is collapsed.
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	loadFromXml	DiagramNode.loadFromXml override. Loads the node's content from an XML element.
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	redimTable	Changes the number of columns and rows in this table.
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	resizeToFitText	Resizes the columns and rows so that cells are large enough to fit their text.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	restoreState	DiagramNode.restoreState override.
	rowFromPoint	Returns the index of the table's row that contains the specified point.
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveState	DiagramNode.saveState override.
	saveToXml	Saves the node's content into an XML element.
	scrollDown	Scrolls the table's rows down.
	scrollUp	Scrolls the table's rows up.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)

	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	toJson	DiagramNode.toJson override.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object. (Inherited from DiagramNode.)
	LoadFromXml	DiagramItem.LoadFromXml override. (Inherited from DiagramNode.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)

	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from DiagramNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)

	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

TableNode Class
MindFusion.Diagramming Namespace

TableNode Fields

TableNode.useScrollBars Field

Set this flag to show full scrollbars for scrollable tables instead of up/down buttons in captions.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
var useScrollBars
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode Constructor

Initializes a new instance of the TableNode class with default values supplied from the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function TableNode ([parent])
```

Parameters

parent
Optional.
Diagram. The Diagram from which to obtain default values for the node properties.

See Also

TableNode Class
MindFusion.Diagramming Namespace

TableNode Methods

TableNode.addColumn Method

Adds a new column to the table.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function addColumn ()
```

Return Value

Object. The newly created column object.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
deleteColumn Method
addRow Method
redimTable Method

TableNode.addRow Method

Adds a new row to the table.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function addRow ()
```

Return Value

Object. The newly created row object.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
deleteRow Method
addColumn Method
redimTable Method

TableNode.cellFromPoint Method

Returns the cell that contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function cellFromPoint (point)
```

Parameters

point
A Point instance.

Return Value

A JavaScript object containing following fields:
{
cell: Cell reference,
cellRect: Rect,
column: Number,
row: Number
};

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.clone Method

DiagramNode.clone override.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function clone ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.containsPoint Method

Checks if the TableNode contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function containsPoint (point)
```

Parameters

point
The Point to check.

Return Value

true if the point is within the bounds of this node, or **false** otherwise.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.createConnectionPoint Method

Creates a connection point between this node and the specified link at the specified point.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function createConnectionPoint (link, point, incoming)
```

Parameters

link
DiagramLink. A DiagramLink that should be connected to this node.

point
Point. The point at which the link should be connected.

incoming
Boolean. true if link is an incoming link, otherwise, false.

Return Value

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.deleteColumn Method

Deletes the specified column.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function deleteColumn (col)
```

Parameters

col
Number. An integer value specifying which column to delete.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.deleteRow Method

Deletes the specified row.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function deleteRow (row)
```

Parameters

row
Number. An integer value specifying which row to delete.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.fromJson Method

DiagramNode.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function fromJson ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.getCell Method

Returns a reference to the cell located at the specified column and row of this table.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function getCell (col, row)
```

Parameters

col
Number. An integer value specifying the index of a table's column.

row
Number. An integer value specifying the index of a table's row.

Return Value

A Cell instance representing the specified cell.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.getColumn Method

Returns a reference to the column at the specified position within the table's list of columns.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function getColumn (col)
```

Parameters

col
Number. An integer value specifying the index of a table's column.

Return Value

A TableColumn instance.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
columnCount Property

TableNode.getOutline Method

Gets an array of points that define the outline of this node shape.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function getOutline ()
```

Return Value

Array. An array with the points.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.getParentRow Method

Returns the number of the first header row before thespecified row. If the specified row is a header rowitself, returns 'row'. If there is no header rowbefore this one, the return value is -1.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function getParentRow ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.getRow Method

Returns a reference to the row at the specified position within the table's list of rows.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function getRow (row)
```

Parameters

row
Number. An integer value specifying the index of a table's row.

Return Value

A TableRow instance.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
rowCount Property

TableNode.getRowRect Method

Returns the bounding rectangle of the specified row.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function getRowRect (row)
```

Parameters

row
Number. An integer value specifying the index of a table's row.

Return Value

Rect. A Rect object representing the row's bounding rectangle.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.insertColumn Method

Inserts a new column at the specified position.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function insertColumn (col)
```

Parameters

col
Number. An integer value specifying the position within the table's list of columns where the new column should be inserted.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
columnCount Property

TableNode.insertRow Method

Inserts a new row at the specified position.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function insertRow (row)
```

Parameters

row
Number. An integer value specifying the position within the table's list of rows where the new row should be inserted.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
rowCount Property

TableNode.isRowCollapsed Method

Checks whether the specified row is collapsed.Collapsed are all rows whose header row is collapsed.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function isRowCollapsed ()
```

See Also

TableNode Members
TableNode Class

TableNode.loadFromXml Method

DiagramNode.loadFromXml override. Loads the node's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.redimTable Method

Changes the number of columns and rows in this table.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function redimTable (columns, rows)
```

Parameters

columns
Number. An integer value specifying the new number of columns.

rows
Number. An integer value specifying the new number of rows.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.resizeToFitText Method

Resizes the columns and rows so that cells are large enough to fit their text.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function resizeToFitText (ignoreCaption, keepCellWidth)
```

Parameters

ignoreCaption
true to ignore caption text and consider only cells for table size, or **false** otherwise.

keepCellWidth
true to preserve column widths and resize only row heights, or **false** otherwise.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.restoreState Method

DiagramNode.restoreState override.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function restoreState ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.rowFromPoint Method

Returns the index of the table's row that contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function rowFromPoint (point)
```

Parameters

point
Point. A Point instance.

Return Value

Number. An integer value specifying the row's index.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.saveState Method

DiagramNode.saveState override.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function saveState ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.saveToXml Method

Saves the node's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.scrollDown Method

Scrolls the table's rows down.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript — Copy Code

```
function scrollDown ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
currentScrollRow Property

TableNode.scrollUp Method

Scrolls the table's rows up.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function scrollUp ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
currentScrollRow Property

TableNode.toJson Method

DiagramNode.toJson override.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode Properties

TableNode.allowResizeColumns Property

Gets or sets a value indicating whether users are allowed to resize table columns.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get allowResizeColumns() {}
set allowResizeColumns(value) {}
```

Property Value

Boolean

true if users are allowed to resize table columns, or **false** otherwise.

Remarks

A column can be resized by dragging its right border. If the columnStyle field of a column is set to AutoWidth, it is automatically reset to **FixedWidth** when the column is resized. Rightmost columns always fill all remaining space in their tables and cannot be

resized.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
tableColumnResized Event
tableColumnResizing Event

TableNode.allowResizeRows Property

Gets or sets a value indicating whether users are allowed to resize table rows.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get allowResizeRows() {}
set allowResizeRows(value) {}
```

Property Value

Boolean

true if users are allowed to resize table rows, or **false** otherwise.

Remarks

A row can be resized by dragging its bottom border.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
tableRowResized Event
tableRowResizing Event

TableNode.captionBackBrush Property

Gets or sets an object specifying how to fill the caption bar.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get captionBackBrush() {}
set captionBackBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.captionFont Property

Gets or sets the font used to render the node's caption text.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get captionFont() {}
set captionFont(value) {}
```

Property Value

Font

A Font object specifying the font of the caption bar.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.captionHeight Property

Gets or sets the height of the node's caption area.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get captionHeight() {}
set captionHeight(value) {}
```

Property Value

Number

A number specifying the height of the caption bar. The default value is 6.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.cellFrameStyle Property

Gets or sets the style of cells' frame lines.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get cellFrameStyle() {}
set cellFrameStyle(value) {}
```

Property Value

CellFrameStyle

A member of the CellFrameStyle enumeration.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.cellStyle Property

Gets or sets the style used to draw table's cells.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get cellStyle() {}
set cellStyle(value) {}
```

Property Value

Style

An instance of the Style class.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.cellTextColor Property

Gets or sets the color used to fill cells text.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get cellTextColor() {}
set cellTextColor(value) {}
```

Property Value

String

A string value identifying the text fill color.

Remarks

If this property is not set, cells derive their text color from table's textColor property, which also applies to the table's caption. You can also set textColor property of individual cells.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.columnCount Property

Gets or sets the number of columns in the table.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get columnCount() {}
set columnCount(value) {}
```

Property Value

Number

The number of columns in the table.

Remarks

If you increase the number of columns the content of the initialcolumns will remain intact. If you decrease the number, the lastcolumns are removed and the other remain unchanged.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.connectionStyle Property

Gets or sets whether drawing a link from/to this node should connectan individual table row, or the whole table as an integral entity.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get connectionStyle() {}
set connectionStyle(value) {}
```

Property Value

ConnectionStyle

A member of the ConnectionStyle enumeration.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.currentScrollRow Property

Gets or sets the table's current scroll position.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get currentScrollRow() {}
set currentScrollRow(value) {}
```

Property Value

Number

The index of the first row that should be visible in the table.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.enableStyledText Property

Gets or sets a value indicating whether styled text rendering is enabled.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get enableStyledText() {}
set enableStyledText(value) {}
```

Property Value

Boolean

true to parse the node's text in order to find formatting tags, or **false** otherwise.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.rowAnchorPattern Property

Gets or sets the default anchor pattern for the rows of a table.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

JavaScript Copy Code

```
get rowAnchorPattern() {}
set rowAnchorPattern(value) {}
```

Property Value

AnchorPattern

An AnchorPattern object that defines a set of anchor points.

Remarks

You can specify individual row's anchor points by setting the anchorPattern property of TableRow.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.rowCount Property

Gets or sets the number of rows in the table.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

```
JavaScript
get rowCount() {}
set rowCount(value) {}
```

Property Value

Number

The number of rows in the table.

Remarks

If you increase the number of rows the content of the initialrows will remain intact. If you decrease the number, the lastrows are removed and the other remain unchanged.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.scrollable Property

Gets or sets whether the user is allowed to scroll the table rows.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

```
JavaScript
get scrollable() {}
set scrollable(value) {}
```

Property Value

Boolean

true to allow scrolling, or **false** otherwise.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.shape Property

Gets or sets the shape of the node's outline.

Namespace: MindFusion.Diagramming
Assembly: TableNode.js

Syntax

```
JavaScript
get shape() {}
set shape(value) {}
```

Property Value

SimpleShape

A member of the SimpleShape enumeration.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
cornerRadius Property

TableRow Class

Represents a row of a TableNode.

Syntax

JavaScript Copy Code

```
// class
TableRow.prototype = {}
```

Remarks

Call the getRow method of TableNode to get the **TableRow** instance at specified position. In order to add or remove rows, set table's rowCount property, or call methods such as addRow, insertRow, deleteRow, redimTable.

Inheritance Hierarchy

MindFusion.Diagramming.TableRow

See Also

TableRow Members
MindFusion.Diagramming Namespace
TableColumn Class

TableRow Members

Represents a row of a TableNode.

The following tables list the members exposed by the TableRow type.

Public Constructors

	Name	Description
	TableRow	Initializes a new instance of the TableRow class.

Public Properties

	Name	Description
	anchorPattern	Gets or sets the anchor points to which links attach when connected to this row.
	expanded	Gets or sets a value indicating whether the table section starting from this header row is collapsed or expanded.
	header	Gets or sets a value indicating whether this row is a section header.
	height	Gets or sets the height of this row.

See Also

TableRow Class
MindFusion.Diagramming Namespace

TableRow Constructor

Initializes a new instance of the TableRow class.

Namespace: MindFusion.Diagramming
Assembly: TableRow.js

Syntax

JavaScript — Copy Code

```
function TableRow ()
```

See Also

TableRow Class
MindFusion.Diagramming Namespace

TableRow Properties

TableRow.anchorPattern Property

Gets or sets the anchor points to which links attach when connected to this row.

Namespace: MindFusion.Diagramming
Assembly: TableRow.js

Syntax

JavaScript — Copy Code

```
get anchorPattern() {}
set anchorPattern(value) {}
```

Property Value

AnchorPattern

An AnchorPattern object that defines a set of anchor points.

Remarks

If this property is not set, links attach to points defined by table's rowAnchorPattern property.

See Also

TableRow Members
TableRow Class
MindFusion.Diagramming Namespace

TableRow.expanded Property

Gets or sets a value indicating whether the table section starting from this header row is collapsed or expanded.

Namespace: MindFusion.Diagramming
Assembly: TableRow.js

Syntax

JavaScript — Copy Code

```
get expanded() {}
set expanded(value) {}
```

Property Value

Boolean

true if the table section that starts from this row is expanded; or **false** otherwise. The default is **true**.

Remarks

This property can be set only if the row is a header. Links connected to the rows of a collapsed table section appear to be connected to the section header row.

See Also

TableRow Members
TableRow Class
MindFusion.Diagramming Namespace

TableRow.header Property

Gets or sets a value indicating whether this row is a section header.

Namespace: MindFusion.Diagramming
Assembly: TableRow.js

Syntax

JavaScript Copy Code

```
get header() {}
set header(value) {}
```

Property Value

Boolean

true if this is the header row of a table section, otherwise **false**. The default is **false**.

Remarks

Each header row defines a section of the table that includes all non-header rows after the header. Such sections can be expanded or collapsed, that is, the rows included in them can be shown or hidden. This can be done interactively by pressing the [±] button displayed in header rows, or programmatically by setting the expanded property. The tableSectionCollapsed and tableSectionExpanded events are raised when a user clicks the [±] button.

See Also

TableRow Members
TableRow Class
MindFusion.Diagramming Namespace

TableRow.height Property

Gets or sets the height of this row.

Namespace: MindFusion.Diagramming
Assembly: TableRow.js

Syntax

JavaScript Copy Code

```
get height() {}
set height(value) {}
```

Property Value

Number

A number specifying the width.

See Also

Theme Class

Represents a set of Style objects whose properties are used to render items of a specific type in the current diagram.

Syntax

JavaScript Copy Code

```
// class
Theme.prototype = {}
```

Remarks

Theme objects can be assigned to the theme property of Diagram.

Inheritance Hierarchy

MindFusion.Diagramming.Theme

See Also

Theme Members
MindFusion.Diagramming Namespace

Theme Members

Represents a set of Style objects whose properties are used to render items of a specific type in the current diagram.

The following tables list the members exposed by the Theme type.

Public Constructors

	Name	Description
	Theme	Initializes a new instance of the Theme class.

Public Properties

	Name	Description
	styles	A dictionary object that maps item type identifiers to **Style** objects.

Public Methods

	Name	Description
	fromId	Gets a built-in theme by its string identificator.
	getBlack	Gets the built-in Black theme.
	getBlue	Gets the built-in Blue theme.
	getBusiness	Gets the built-in Business theme.
	getEarth	Gets the built-in Earth theme.
	getGray	Gets the built-in Gray theme.
	getGreen	Gets the built-in Green theme.
	getLight	Gets the built-in Light theme.

	getPastel	Gets the built-in Pastel theme.
	getPeach	Gets the built-in Peach theme.
	getStandard	Gets the built-in Standard theme.

See Also

Theme Class
MindFusion.Diagramming Namespace

Theme Constructor

Initializes a new instance of the Theme class.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript Copy Code

```
function Theme ()
```

See Also

Theme Class
MindFusion.Diagramming Namespace

Theme Methods

Theme.fromId Method

Gets a built-in theme by its string identificator.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript Copy Code

```
function fromId (id)
```

Parameters

id
String. The theme identificator.

Return Value

Theme. An instance of the Theme class.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.getBlack Method

Gets the built-in Black theme.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript Copy Code

```
function getBlack ()
```

Return Value

Theme. An instance of the Theme class.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.getBlue Method

Gets the built-in Blue theme.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript Copy Code

```
function getBlue ()
```

Return Value

An instance of the Theme class.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.getBusiness Method

Gets the built-in Business theme.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript Copy Code

```
function getBusiness ()
```

Return Value

An instance of the Theme class.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.getEarth Method

Gets the built-in Earth theme.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript Copy Code

```
function getEarth ()
```

Return Value

An instance of the Theme class.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.getGray Method

Gets the built-in Gray theme.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript Copy Code

```
function getGray ()
```

Return Value

An instance of the Theme class.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.getGreen Method

Gets the built-in Green theme.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript Copy Code

```
function getGreen ()
```

Return Value

An instance of the Theme class.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.getLight Method

Gets the built-in Light theme.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript — Copy Code

```
function getLight ()
```

Return Value

An instance of the Theme class.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.getPastel Method

Gets the built-in Pastel theme.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript — Copy Code

```
function getPastel ()
```

Return Value

An instance of the Theme class.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.getPeach Method

Gets the built-in Peach theme.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript — Copy Code

```
function getPeach ()
```

Return Value

An instance of the Theme class.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.getStandard Method

Gets the built-in Standard theme.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript — Copy Code

```
function getStandard ()
```

Return Value

An instance of the Theme class.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme Properties

Theme.styles Property

A dictionary object that maps item type identifiers to Style objects.

Namespace: MindFusion.Diagramming
Assembly: Theme.js

Syntax

JavaScript — Copy Code

```
get styles() {}
set styles(value) {}
```

Property Value

Map

Maps item types to Style objects.

Example

The following code sets default white background of ShapeNode objects when the theme is applied.

JavaScript — Copy Code

```
var shapeNodeStyle = new Style();
shapeNodeStyle.brush = "white";
theme.styles.set("std:ShapeNode", shapeNodeStyle);
```

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

TreeItemEventArgs Class

Contains the arguments passed to handlers of tree view items -related events.

Syntax

```
// class
TreeItemEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Controls.CancelEventArgs
  MindFusion.Diagramming.TreeItemEventArgs

See Also

TreeItemEventArgs Members
MindFusion.Diagramming Namespace
Events Class

TreeItemEventArgs Members

Contains the arguments passed to handlers of tree view items -related events.

The following tables list the members exposed by the TreeItemEventArgs type.

Public Constructors

	Name	Description
	TreeItemEventArgs	Initializes a new instance of the TreeItemEventArgs class.

Public Properties

	Name	Description
	cancel	Gets or sets a value indicating whether to allow the current operation. (Inherited from CancelEventArgs.)
	context	Gets the canvas rendering context.
	handled	Gets or sets a value indicating whether the event has been handled. (Inherited from EventArgs.)
	item	Gets the item related to the event.
	mouseButton	Gets which mouse button has been pressed.
	newText	Gets the text that has just been entered by the user.
	node	Gets the node related to the event.
	oldText	Gets the text that had been displayed in the node before the user edited it.

See Also

TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemEventArgs Constructor

Initializes a new instance of the TreeItemEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function TreeItemEventArgs ()
```

See Also

TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemEventArgs Properties

TreeItemEventArgs.context Property

Gets the canvas rendering context.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get context() {}
set context(value) {}
```

Property Value

CanvasRenderingContext2D

A rendering context used to draw on the Canvas element.

See Also

TreeItemEventArgs Members
TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemEventArgs.item Property

Gets the item related to the event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get item() {}
set item(value) {}
```

Property Value

TreeViewItem

A TreeViewItem instance.

See Also

TreeItemEventArgs Members
TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemEventArgs.mouseButton Property

Gets which mouse button has been pressed.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get mouseButton() {}
set mouseButton(value) {}
```

Property Value

Number

An integer mouse button id.

See Also

TreeItemEventArgs Members
TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemEventArgs.newText Property

Gets the text that has just been entered by the user.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get newText() {}
set newText(value) {}
```

Property Value

String

A string containing newly entered text.

See Also

TreeItemEventArgs Members
TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemEventArgs.node Property

Gets the node related to the event.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get node() {}
set node(value) {}
```

Property Value

TreeViewNode

A TreeViewNode instance.

See Also

TreeItemEventArgs Members
TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemEventArgs.oldText Property

Gets the text that had been displayed in the node before the user edited it.

Namespace: MindFusion.Diagramming
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get oldText() {}
set oldText(value) {}
```

Property Value

String

A string containing item's original text.

See Also

TreeItemEventArgs Members
TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint Class

A class that represents a connection between an edge and a node in a graph.

Syntax

JavaScript Copy Code

```
// class
TreeViewConnectionPoint.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.ConnectionPoint
MindFusion.Diagramming.TreeViewConnectionPoint

See Also

TreeViewConnectionPoint Members
MindFusion.Diagramming Namespace

TreeViewConnectionPoint Members

A class that represents a connection between an edge and a node in a graph.

The following tables list the members exposed by the TreeViewConnectionPoint type.

Public Properties

	Name	Description
	endPoint	Calculates the coordinates of this connection point expressed in document coordinates.
	RelativePosition	Gets or sets the position of this connection point, relative to the associated node's bounding rectangle, expressed as percentage. (Inherited from

		ConnectionPoint.)
	Row	Gets the index of a constituent item within the associated node where the link represented by this connection point is connected, or -1, if the link is connected to the node itself. (Inherited from ConnectionPoint.)

Protected Properties

	Name	Description
	Incoming	Gets a flag indicating whether this connection point is at the origin or the destination of the associated link. (Inherited from ConnectionPoint.)
	Link	Gets the link associated with this connection point. (Inherited from ConnectionPoint.)
	Node	Gets the node associated with this connection point. (Inherited from ConnectionPoint.)

Public Methods

	Name	Description
	addLinkToNode	Adds the link associated with this connection point to the associated node.
	AddLinkToNode	Adds the link associated with this connection point to the associated node. (Inherited from ConnectionPoint.)
	CalculateIntersections	Determines whether the end of a link should be aligned to the point where the node contour is intersected by the line connecting the centers of the connected nodes. (Inherited from ConnectionPoint.)
	CanMoveLink	Checks whether the specified point, expressed in document coordinates, is a valid location for this connection point. (Inherited from ConnectionPoint.)
	GetAnchorPos	Overloaded. (Inherited from ConnectionPoint.)
	GetEndPoint	Calculates the coordinates of the this connection point expressed in document coordinates. (Inherited from ConnectionPoint.)
	getInitialPoint	Returns a point (usually the center of the bounding rectangle) contained within the associated node.
	GetInitialPoint	Returns a point (usually the center of the bounding rectangle) contained within the associated node. (Inherited from ConnectionPoint.)
	getIntersection	Calculates the intersection point between the specified line segment and the contour of the node associated with this connection point.
	GetIntersection	Calculates the intersection point between the specified line segment and the contour of the node associated with this connection point. (Inherited from ConnectionPoint.)
	getNearestBorderPoint	Returns the nearest point lying on the node border.
	GetNodeRect	Returns the bounding rectangle of the node associated with this connection point. (Inherited from ConnectionPoint.)
	linkChanges	Checks whether the link would change if attached to the specified node at the specified point.
	LinkChanges	Checks whether the relationship represented by the associated link would change if the link is attached to the specified node at the specified point. (Inherited from ConnectionPoint.)
	NearestAnchorPoint	Returns the anchor point closest to the specified point from the node's associated anchor pattern. (Inherited from ConnectionPoint.)
	NodesIntersect	Checks whether the node associated with the current connection point and the node associated with the specified connection point intersect. (Inherited from ConnectionPoint.)

	removeLinkFromNode	Removes the link associated with this connection point from the associated node.
	RemoveLinkFromNode	Removes the link associated with this connection point from the associated node. (Inherited from ConnectionPoint.)
	SameNode	Checks whether the current connection point and the specified connection point are associated with the same node. (Inherited from ConnectionPoint.)
	saveEndRelative	Updates the coordinates of the connection point represented by this link, relative to the bounding rectangle of the associated node.
	SaveEndRelative	Updates the coordinates of the connection point represented by this link, relative to the bounding rectangle of the associated node. (Inherited from ConnectionPoint.)

Protected Methods

	Name	Description
	GetNearestBorderPoint	Returns the nearest point lying on the node border. (Inherited from ConnectionPoint.)
	UpdateFromPoint	Updates the connection state according to the specified point, without changing the related node. (Inherited from ConnectionPoint.)

See Also

TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint Methods

TreeViewConnectionPoint.addLinkToNode Method

Adds the link associated with this connection point to the associated node.

Namespace: MindFusion.Diagramming
Assembly: TreeViewConnectionPoint.js

Syntax

JavaScript Copy Code

```
function addLinkToNode ()
```

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.getInitialPoint Method

Returns a point (usually the center of the bounding rectangle) contained within the associated node.

Namespace: MindFusion.Diagramming
Assembly: TreeViewConnectionPoint.js

Syntax

JavaScript Copy Code

```
function getInitialPoint ()
```

Return Value

Point. A Point contained within the node.

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.getIntersection Method

Calculates the intersection point between the specified line segment and the contour of the node associated with this connection point.

Namespace: MindFusion.Diagramming
Assembly: TreeViewConnectionPoint.js

Syntax

JavaScript Copy Code

```
function getIntersection (point1, point2)
```

Parameters

point1
Point. The first point of the segment.

point2
Point. The second point of the segment.

Return Value

Point. A Point object representing the intersection point.

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.getNearestBorderPoint Method

Returns the nearest point lying on the node border.

Namespace: MindFusion.Diagramming
Assembly: TreeViewConnectionPoint.js

Syntax

JavaScript Copy Code

```
function getNearestBorderPoint (point)
```

Parameters

point
Point. A Point instance specifying location in the diagram.

Return Value

AnchorPointDetails. An AnchorPointDetails specifying the anchor point properties.

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.linkChanges Method

Checks whether the link would change if attached to the specified node at the specified point.

Namespace: MindFusion.Diagramming
Assembly: TreeViewConnectionPoint.js

Syntax

JavaScript Copy Code

```
function linkChanges (item, point)
```

Parameters

item
DiagramItem. A DiagramItem under the mouse cursor.

point
Point. A point where the link will be connected.

Return Value

Boolean. true if the link would connect to a different entity if dropped at the specified location; otherwise, false.

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.removeLinkFromNode Method

Removes the link associated with this connection point from the associated node.

Namespace: MindFusion.Diagramming
Assembly: TreeViewConnectionPoint.js

Syntax

JavaScript Copy Code

```
function removeLinkFromNode ()
```

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.saveEndRelative Method

Updates the coordinates of the connection point represented by this link, relative to the bounding rectangle of the associated node.

Namespace: MindFusion.Diagramming
Assembly: TreeViewConnectionPoint.js

Syntax

JavaScript Copy Code

```
function saveEndRelative ()
```

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint Properties

TreeViewConnectionPoint.endPoint Property

Calculates the coordinates of this connection point expressed in document coordinates.

Namespace: MindFusion.Diagramming
Assembly: TreeViewConnectionPoint.js

Syntax

JavaScript Copy Code

```
get endPoint() {}
set endPoint(value) {}
```

Property Value

Point

A Point that represents the coordinate.

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewItem Class

Represents a constituent item of the hierarchy displayed inside a TreeViewNode.

Syntax

JavaScript Copy Code

```
// class
TreeViewItem.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.TreeViewItem

See Also

TreeViewItem Members
MindFusion.Diagramming Namespace

TreeViewItem Members

Represents a constituent item of the hierarchy displayed inside a TreeViewNode.

The following tables list the members exposed by the TreeViewItem type.

Public Constructors

	Name	Description
	TreeViewItem	Initializes a new instance of the TreeViewItem class.

Public Properties

	Name	Description
	brush	Gets or sets the brush used to paint item's background.
	children	Gets a list of this item's children.
	expanded	Gets or sets a value indicating whether this item is expanded to show its children.
	font	Gets or sets the font used to render item's text.
	height	Gets or sets the height of this item.
	image	Gets or sets the image displayed by this item.
	imageContent	Gets or sets a Base64-encoded image to display inside this item.
	imageLocation	Gets or sets URL of the image to display inside this item.
	imagePadding	Gets or sets the padding distance between item's borders and image.
	parent	Gets the parent item of this item.
	tag	Gets or sets custom data tag associated with this item.
	text	Gets or sets the text label displayed by this item.
	textColor	Gets or sets the item's text color.
	tooltip	Gets or sets the tooltip text that should be displayed when the mouse hovers over this item.

Public Methods

	Name	Description
	loadFromXml	Loads the item's content from an XML element.
	saveToXml	Saves the item's content into an XML element.

See Also

TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem Constructor

Initializes a new instance of the TreeViewItem class.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
function TreeViewItem ()
```

See Also

TreeViewItem Methods

TreeViewItem.loadFromXml Method

Loads the item's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.saveToXml Method

Saves the item's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem Properties

TreeViewItem.brush Property

Gets or sets the brush used to paint item's background.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get brush() {}
set brush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.children Property

Gets a list of this item's children.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get children() {}
```

Property Value

Array

An array of TreeViewItem objects.

Remarks

Do not add items to this array directly, but call the addItem method of TreeViewNode.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.expanded Property

Gets or sets a value indicating whether this item is expanded to show its children.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get expanded() {}
set expanded(value) {}
```

Property Value

Boolean

true if the item is expanded, or **false** otherwise.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.font Property

Gets or sets the font used to render item's text.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get font() {}
set font(value) {}
```

Property Value

Font

An instance of the Font class.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.height Property

Gets or sets the height of this item.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get height() {}
set height(value) {}
```

Property Value

Number

A number specifying item's height.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.image Property

Gets or sets the image displayed by this item.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get image() {}
set image(value) {}
```

Property Value

HTMLImageElement

An image object to render as item's content.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.imageContent Property

Gets or sets a Base64-encoded image to display inside this item.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get imageContent() {}
set imageContent(value) {}
```

Property Value

String

A Base64-encoded representation of an image.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.imageLocation Property

Gets or sets URL of the image to display inside this item.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get imageLocation() {}
set imageLocation(value) {}
```

Property Value

String

A URL string that identifies the image to be loaded.

See Also

TreeViewItem.imagePadding Property

Gets or sets the padding distance between item's borders and image.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get imagePadding() {}
set imagePadding(value) {}
```

Property Value

Thickness

A Thickness instance specifying image padding.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.parent Property

Gets the parent item of this item.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get parent() {}
set parent(value) {}
```

Property Value

TreeViewItem

The parent item.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.tag Property

Gets or sets custom data tag associated with this item.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get tag() {}
set tag(value) {}
```

Property Value

Object

An Object containing custom data.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.text Property

Gets or sets the text label displayed by this item.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get text() {}
set text(value) {}
```

Property Value

String

A string containing the item's text. The default value is empty string "".

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.textColor Property

Gets or sets the item's text color.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

A string value identifying the text color.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.tooltip Property

Gets or sets the tooltip text that should be displayed when the mouse hovers over this item.

Namespace: MindFusion.Diagramming
Assembly: TreeViewItem.js

Syntax

JavaScript — Copy Code

```
get tooltip() {}
set tooltip(value) {}
```

Property Value

String

A string containing the tooltip text.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewNode Class

TreeViewNode instances are diagram nodes that can draw hierarchical data.

Syntax

JavaScript — Copy Code

```
// class
TreeViewNode.prototype = {}
```

Remarks

The root items displayed in the node can be accessed through the rootItems property. Items can be added and removed individually by using the addItem and removeItem methods, or in bulk by calling the fromObject method, which loads the tree view items from an array of objects.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.CaptionedNode
      MindFusion.Diagramming.TreeViewNode

See Also

TreeViewNode Members
MindFusion.Diagramming Namespace

TreeViewNode Members

TreeViewNode instances are diagram nodes that can draw hierarchical data.

The following tables list the members exposed by the **TreeViewNode** type.

Public Constructors

	Name	Description
	TreeViewNode	Initializes a new instance of the TreeViewNode class with default values supplied from the specified Diagram.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	captionBackBrush	Gets or sets an object specifying how to fill the caption bar.
	captionFont	Gets or sets the font used to render the node's caption text.
	captionHeight	Gets or sets the height of the node's caption area.
	connectionStyle	Gets or sets whether drawing a link from/to this node should connectan individual TreeViewItem, or the whole node as an integral entity.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	dividerStroke	Gets or sets the color used to stroke the divider line between node's caption bar and content area. (Inherited from CaptionedNode.)
	dividerStrokeDashArray	Gets or sets an array of values specifying the dash pattern of the node's divider line. (Inherited from CaptionedNode.)
	dividerStrokeDashOffset	Gets or sets the the distance between the line start and the beginning of the dash pattern. (Inherited from CaptionedNode.)
	dividerStrokeDashStyle	Gets or sets the line dash pattern applied when stroking node's divider line. (Inherited from CaptionedNode.)
	dividerStrokeThickness	Gets or sets the width of the divider line between node's caption bar and content area. (Inherited from CaptionedNode.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	effectiveDividerStrokeThickness	Gets the effective line thickness that should be applied when stroking this node's divider line. (Inherited from CaptionedNode.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)

	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	enableStyledText	Gets or sets a value indicating whether styled text rendering is enabled.
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	flatItems	Gets a list of all items in this node.
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	rootItems	Gets a list of root items displayed in this node.

	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	scrollable	Gets or sets whether the node's items can be scrolled vertically.
	scrollPosition	Gets or sets the node's current scroll position.
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	shape	Gets or sets the shape of the node's outline.
	showLines	Gets or sets whether to display lines connecting child items to parent items.
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description

	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	addItem	Adds a new child item to the specified parent item or to the root items list, if a parent is not specified.
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	clone	DiagramNode.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	containsPoint	Checks if the TreeViewNode contains the specified point.
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	createConnectionPoint	Creates a connection point between this node and the specified link at the specified point.
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	fromJson	DiagramNode.fromJson override.
	fromObject	Loads the tree view items from a data object.
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)

	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	getItemRect	Returns the bounding rectangle of the specified item.
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	getOutline	Gets an array of points that define the outline of this node shape.
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	itemFromPoint	Returns the item that contains the specified point.
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	loadFromXml	DiagramNode.LoadFromXml override. Loads the node's content from an XML element.
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)

	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	removeItem	Removes the specified item from the node.
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	resizeToFitText	Resizes the tree view so that it is large enough to fit the text of its items.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	restoreState	DiagramItem.restoreState override. (Inherited from CaptionedNode.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveState	DiagramItem.saveState override. (Inherited from CaptionedNode.)
	saveToXml	Saves the node's content into an XML element.
	scrollDown	Scrolls the node down.
	scrollUp	Scrolls the table's rows up.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	toJson	DiagramNode.toJson override.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)

	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object. (Inherited from DiagramNode.)
	LoadFromXml	DiagramItem.LoadFromXml override. (Inherited from DiagramNode.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)

	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from DiagramNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

TreeViewNode Constructor

Initializes a new instance of the TreeViewNode class with default values supplied from the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function TreeViewNode ([parent])
```

Parameters

parent
Optional.
Diagram. The Diagram from which to obtain default values for the node properties.

See Also

TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode Methods

TreeViewNode.addItem Method

Adds a new child item to the specified parent item or to the root items list, if a parent is not specified.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function addItem (item, [parentItem])
```

Parameters

item
TreeViewItem. The item to add.

parentItem
Optional.
TreeViewItem. The parent item.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.clone Method

DiagramNode.clone override.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function clone ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.containsPoint Method

Checks if the TreeViewNode contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function containsPoint (point)
```

Parameters

point
Point. The Point to check.

Return Value

Boolean. true if the point is within the bounds of this node; otherwise, false.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.createConnectionPoint Method

Creates a connection point between this node and the specified link at the specified point.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function createConnectionPoint (link, point, incoming)
```

Parameters

link
DiagramLink. A DiagramLink that should be connected to this node.

point
Point. The point at which the link should be connected.

incoming
Boolean. true if link is an incoming link, otherwise, false.

Return Value

TreeViewConnectionPoint. The connection point.

See Also

TreeViewNode.fromJson Method

DiagramNode.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function fromJson ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.fromObject Method

Loads the tree view items from a data object.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function fromObject (data)
```

Parameters

data
Object. The object containing the tree view data.

Example

JavaScript Copy Code

```
node.fromObject(
[{
    text: "Order",
    imageLocation: "icon_calculator.png"
},
{
    text: "Date",
    imageLocation: "icon_time.png"
},
{
    text: "Products",
    imageLocation: "icon_delivery.png",
    children: [{ text: "Name" }, { text: "TotalPrice", brush: "lightblue" }]
}]
);
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.getItemRect Method

Returns the bounding rectangle of the specified item.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function getItemRect (item)
```

Parameters

item
TreeViewItem. The item.

Return Value

Rect. A Rect object representing the item's bounding rectangle.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.getOutline Method

Gets an array of points that define the outline of this node shape.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function getOutline ()
```

Return Value

Array. An array with the points.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.itemFromPoint Method

Returns the item that contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function itemFromPoint (point)
```

Parameters

point
Point. A Point instance.

Return Value

TreeViewItem. The item containing the point, or null.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.loadFromXml Method

DiagramNode.LoadFromXml override. Loads the node's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.removeItem Method

Removes the specified item from the node.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function removeItem (item)
```

Parameters

item
TreeViewItem. The item to remove.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.resizeToFitText Method

Resizes the tree view so that it is large enough to fit the text of its items.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function resizeToFitText ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.saveToXml Method

Saves the node's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.scrollDown Method

Scrolls the node down.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function scrollDown ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.scrollUp Method

Scrolls the table's rows up.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function scrollUp ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.toJson Method

DiagramNode.toJson override.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode Properties

TreeViewNode.captionBackBrush Property

Gets or sets an object specifying how to fill the caption bar.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
get captionBackBrush() {}
set captionBackBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.captionFont Property

Gets or sets the font used to render the node's caption text.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
get captionFont() {}
set captionFont(value) {}
```

Property Value

Font

An object specifying the font of the caption bar.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.captionHeight Property

Gets or sets the height of the node's caption area.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
get captionHeight() {}
set captionHeight(value) {}
```

Property Value

Number

A number specifying the height of the caption bar. The default value is 6.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.connectionStyle Property

Gets or sets whether drawing a link from/to this node should connectan individual TreeViewItem, or the whole node as an integral entity.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
get connectionStyle() {}
set connectionStyle(value) {}
```

Property Value

ConnectionStyle

A member of the ConnectionStyle enumeration.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.enableStyledText Property

Gets or sets a value indicating whether styled text rendering is enabled.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
get enableStyledText() {}
set enableStyledText(value) {}
```

Property Value

Boolean

true to parse the node's text in order to find formatting tags, or **false** otherwise.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.flatItems Property

Gets a list of all items in this node.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
get flatItems() {}
set flatItems(value) {}
```

Property Value

Array

An array containing all TreeViewItem objects in the node.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.rootItems Property

Gets a list of root items displayed in this node.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
get rootItems() {}
set rootItems(value) {}
```

Property Value

Array

An array of root TreeViewItem objects.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.scrollable Property

Gets or sets whether the node's items can be scrolled vertically.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
get scrollable() {}
set scrollable(value) {}
```

Property Value

Boolean

true to allow scrolling, or **false** otherwise.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.scrollPosition Property

Gets or sets the node's current scroll position.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
get scrollPosition() {}
set scrollPosition(value) {}
```

Property Value

Number

A number specifying the scroll position.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.shape Property

Gets or sets the shape of the node's outline.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
get shape() {}
set shape(value) {}
```

Property Value

SimpleShape

A member of the SimpleShape enumeration.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.showLines Property

Gets or sets whether to display lines connecting child items to parent items.

Namespace: MindFusion.Diagramming
Assembly: TreeViewNode.js

Syntax

JavaScript Copy Code

```
get showLines() {}
set showLines(value) {}
```

Property Value

Boolean

true to display connecting lines, or **false** otherwise.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

UndoEventArgs Class

Contains the arguments passed to the actionUndone and actionRedone event handlers.

Syntax

JavaScript Copy Code

```
// class
UndoEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.EventArgs
MindFusion.Diagramming.UndoEventArgs

See Also

UndoEventArgs Members
MindFusion.Diagramming Namespace

UndoEventArgs Members

Contains the arguments passed to the actionUndone and actionRedone event handlers.

The following tables list the members exposed by the UndoEventArgs type.

Public Constructors

	Name	Description
	UndoEventArgs	Initializes a new instance of the UndoEventArgs class.

Public Properties

	Name	Description
	command	Gets the command related to the event.

See Also

UndoEventArgs Class
MindFusion.Diagramming Namespace

UndoEventArgs Constructor

Initializes a new instance of the UndoEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

JavaScript Copy Code

```
function UndoEventArgs (args)
```

Parameters

args

See Also

UndoEventArgs Class
MindFusion.Diagramming Namespace

UndoEventArgs Properties

UndoEventArgs.command Property

Gets the command related to the event.

Namespace: MindFusion.Diagramming
Assembly: Diagram.js

Syntax

```
JavaScript                                                    Copy Code
get command() {}
set command(value) {}
```

Property Value

Command

A Command instance.

See Also

UndoEventArgs Members
UndoEventArgs Class
MindFusion.Diagramming Namespace

VideoNode Class

A **VideoNode** instance renders video stream assigned to the videoLocation property.

Syntax

```
JavaScript                                                    Copy Code
// class
VideoNode.prototype = {}
```

Remarks

VideoNode objects display video streams, along with UI elements to play, pause, seek and change volume. Set node's videoLocation method to specify its video URL. Auto-playing is not supported at this time, and the video can be started only by users clicking the play button.

VideoNode is built around Video component, which can also be used in CompositeNode templates.

Inheritance Hierarchy

MindFusion.Diagramming.DiagramItem
  MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.VideoNode

See Also

VideoNode Members
MindFusion.Diagramming Namespace

VideoNode Members

A VideoNode instance renders video stream assigned to the videoLocation property.

The following tables list the members exposed by the **VideoNode** type.

Public Constructors

	Name	Description
	VideoNode	Initializes a new instance of the **VideoNode** class.

Protected Fields

	Name	Description

	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)

	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)

	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	videoLocation	Gets or sets the URL of the video stream displayed in this VideoNode.
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	clone	DiagramNode.clone override.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)

DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)
DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
fromJson	DiagramNode.fromJson override.
GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
loadFromXml	DiagramNode.loadFromXml override. Loads the node's content from an XML element.
Move	Moves the node to the specified location. (Inherited from DiagramNode.)

	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	saveToXml	Saves the node's content into an XML element.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	toJson	DiagramNode.toJson override.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)

	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object. (Inherited from DiagramNode.)
	LoadFromXml	DiagramItem.LoadFromXml override. (Inherited from DiagramNode.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)

	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from DiagramNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

VideoNode Constructor

Initializes a new instance of the VideoNode class.

Namespace: MindFusion.Diagramming
Assembly: VideoNode.js

Syntax

JavaScript Copy Code

```
function VideoNode ([parent])
```

Parameters

parent
Optional.
A Diagram instance whose default node attributes are copied to this node.

See Also

VideoNode Class
MindFusion.Diagramming Namespace

VideoNode Methods

VideoNode.clone Method

DiagramNode.clone override.

Namespace: MindFusion.Diagramming
Assembly: VideoNode.js

Syntax

JavaScript Copy Code

```
function clone ()
```

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode.fromJson Method

DiagramNode.fromJson override.

Namespace: MindFusion.Diagramming
Assembly: VideoNode.js

Syntax

JavaScript Copy Code

```
function fromJson ()
```

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode.loadFromXml Method

DiagramNode.loadFromXml override. Loads the node's content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: VideoNode.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element containing the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode.saveToXml Method

Saves the node's content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: VideoNode.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode.toJson Method

DiagramNode.toJson override.

Namespace: MindFusion.Diagramming
Assembly: VideoNode.js

Syntax

JavaScript — Copy Code

```
function toJson ()
```

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode Properties

VideoNode.videoLocation Property

Gets or sets the URL of the video stream displayed in this VideoNode.

Namespace: MindFusion.Diagramming
Assembly: VideoNode.js

Syntax

JavaScript — Copy Code

```
get videoLocation() {}
set videoLocation(value) {}
```

Property Value

String

A string containing video stream URL.

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

WebComponents Class

Implements registration of web components.

Syntax

JavaScript — Copy Code

```
// class
WebComponents.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.WebComponents

See Also

WebComponents Members
MindFusion.Diagramming Namespace

WebComponents Members

Implements registration of web components.

The following tables list the members exposed by the WebComponents type.

Public Methods

	Name	Description
	register	Registers the controls in the package as web components.

See Also

WebComponents Class
MindFusion.Diagramming Namespace

WebComponents Methods

WebComponents.register Method

Registers the controls in the package as web components.

Namespace: MindFusion.Diagramming
Assembly: WebComponents.js

Syntax

JavaScript Copy Code

```
function register ()
```

Remarks

Call this method to register the control classes from Diagramming package as web components. After registration, the browser instantiates corresponding classes for following tags in HTML code:

HTML	JS Class
<mindfusion-diagramview>	DiagramView
<mindfusion-overview>	Overview
<mindfusion-nodelistview>	NodeListView
<mindfusion-ruler>	Ruler
<mindfusion-zoomcontrol>	ZoomControl

You can get a reference to objects created by the browser by calling the find method of corresponding class with **id** attribute's value as argument.

See Also

WebComponents Members
WebComponents Class
MindFusion.Diagramming Namespace
create Method

XmlPersistContext Class

Contains helper methods for serialization of diagram contents to and from XML documents.

Syntax

JavaScript — Copy Code

```
// class
XmlPersistContext.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.XmlPersistContext

See Also

XmlPersistContext Members
MindFusion.Diagramming Namespace

XmlPersistContext Members

Contains helper methods for serialization of diagram contents to and from XML documents.

The following tables list the members exposed by the XmlPersistContext type.

Public Constructors

	Name	Description
	XmlPersistContext	Initializes a new instance of the XmlPersistContext class.

Public Methods

	Name	Description
	addChildElement	Adds a new child node with the specified name and value to the specified parent node.
	readBool	Reads a Boolean value with the specified name.
	readBrush	Reads a Brush object with the specified name.
	readFloat	Reads a floating-point number with the specified name.
	readImage	Reads an image object with the specified name.
	readInt	Reads an integer number with the specified name.
	readString	Reads a string value with the specified name.
	writeBool	Writes a Boolean value with the specified name.
	writeBrush	Writes a reference to the specified brush. The brush is registered within the internal hashtable for subsequent serialization.
	writeFloat	Writes a floating-point number with the specified name.
	writeImage	Writes a reference to the specified image. The image is registered within the internal hashtable for subsequent serialization.
	writeInt	Writes an integer number with the specified name.
	writeString	Writes a string value with the specified name.

See Also

XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext Constructor

Initializes a new instance of the XmlPersistContext class.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript — Copy Code

```
function XmlPersistContext (diagram)
```

Parameters

diagram
Diagram. The Diagram that will be serialized or deserialized.

See Also

XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext Methods

XmlPersistContext.addChildElement Method

Adds a new child node with the specified name and value to the specified parent node.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript — Copy Code

```
function addChildElement (elementName, parentElement, [innerText])
```

Parameters

elementName
String. A string containing the name of new child element.

parentElement
Element. The parent DOM Element object.

innerText
Optional.
String. Optional value to be added as child text node of the new element.

Return Value

Element. The newly created DOM Element object.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.readBool Method

Reads a Boolean value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function readBool (elementName, parentElement, defaultValue)
```

Parameters

elementName
String. A string specifying the name of the Boolean XML element.

parentElement
Element. The parent DOM Element object.

defaultValue
Boolean. The default value to return if specified child XML element does not exist.

Return Value

Boolean. The value loaded from XML.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.readBrush Method

Reads a Brush object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function readBrush (elementName, parentElement, [defaultValue])
```

Parameters

elementName
String. A string specifying the name of the brush XML element.

parentElement
Element. The parent DOM Element object.

defaultValue
Optional.
Object. The default brush to return if specified child XML element does not exist.

Return Value

Object. The brush object loaded from XML.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.readFloat Method

Reads a floating-point number with the specified name.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function readFloat (elementName, parentElement, [defaultValue])
```

Parameters

elementName
String. A string specifying the name of the number XML element.

parentElement
Element. The parent DOM Element object.

defaultValue
Optional.
Number. The default number to return if specified child XML element does not exist.

Return Value

Number. The number loaded from XML.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.readImage Method

Reads an image object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function readImage (elementName, parentElement)
```

Parameters

elementName
String. A string specifying the name of the image XML element.

parentElement
Element. The parent DOM Element object.

Return Value

String. Base64-encoded image data loaded from XML.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.readInt Method

Reads an integer number with the specified name.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function readInt (elementName, parentElement, [defaultValue])
```

Parameters

elementName
String. A string specifying the name of the number XML element.

parentElement
Element. The parent DOM Element object.

defaultValue
Optional.
Number. The default number to return if specified child XML element does not exist.

Return Value

Number. The number loaded from XML.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.readString Method

Reads a string value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function readString (elementName, parentElement, [defaultValue])
```

Parameters

elementName
String. The name of the string's XML element.

parentElement
Element. The parent DOM Element object.

defaultValue
Optional.
Object. The default string to return if specified child XML element does not exist.

Return Value

String. The string loaded from XML.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.writeBool Method

Writes a Boolean value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function writeBool (boolValue, elementName, parentElement)
```

Parameters

boolValue
Boolean. The Boolean value that should be saved in current XML document.

elementName
String. A string specifying the name of the new element.

parentElement
Element. The parent DOM Element object.

Return Value
Element. The newly created DOM Element object.

See Also
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.writeBrush Method

Writes a reference to the specified brush. The brush is registered within the internal hashtable for subsequent serialization.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function writeBrush (brush, elementName, parentElement)
```

Parameters
brush
Object. The Brush object that should be saved in current XML document.

elementName
String. A string specifying the name of the new element.

parentElement
Element. The parent DOM Element object.

Return Value
Element. The newly created DOM Element object.

See Also
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.writeFloat Method

Writes a floating-point number with the specified name.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function writeFloat (floatValue, elementName, parentElement)
```

Parameters
floatValue
Number. The number that should be saved in current XML document.

elementName
String. A string specifying the name of the new element.

parentElement
Element. The parent DOM Element object.

Return Value

Element. The newly created DOM Element object.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.writeImage Method

Writes a reference to the specified image. The image is registered within the internal hashtable for subsequent serialization.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function writeImage (image, elementName, parentElement)
```

Parameters

image
String. Base64-encoded image data that should be saved in XML.

elementName
String. A string specifying the name of the new element.

parentElement
Element. The parent DOM Element object.

Return Value

Element. The newly created DOM Element object.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.writeInt Method

Writes an integer number with the specified name.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function writeInt (intValue, elementName, parentElement)
```

Parameters

intValue
Number. The number that should be saved in current XML document.

elementName
String. A string specifying the name of the new element.

parentElement
Element. The parent DOM Element object.

Return Value

Element. The newly created DOM Element object.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.writeString Method

Writes a string value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: XmlPersistContext.js

Syntax

JavaScript Copy Code

```
function writeString (stringValue, elementName, parentElement)
```

Parameters

stringValue
String. The string that should be saved in current XML document.

elementName
String. A string specifying the name of the new element.

parentElement
Element. The parent DOM Element object.

Return Value

Element. The newly created DOM Element object.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

MindFusion.Diagramming.Animations Namespace

BounceAnimation Class

Animates vertical position of nodes.

Syntax

JavaScript Copy Code

```
// class
BounceAnimation.prototype = {}
```

Remarks

Call item's animate method to run an animation.

Example

JavaScript Copy Code

```
node.animate(
    new BounceAnimation(
        {
            fromValue: 0,
            toValue: -5,
```

```
            duration: 250,
            repeat: true,
            repeatCount: 4,
            reverse: true,
            animationType: Animations.AnimationType.BackBow
        }));
```

Inheritance Hierarchy

MindFusion.Animations.Animation
  MindFusion.Diagramming.Animations.ItemAnimation
    MindFusion.Diagramming.Animations.BounceAnimation

See Also

BounceAnimation Members
MindFusion.Diagramming.Animations Namespace

BounceAnimation Members

Animates vertical position of nodes.

The following tables list the members exposed by the BounceAnimation type.

Public Fields

	Name	Description
	AnimationsRunning	Track how many animations are running. (Inherited from Animation.)
	InvokeOnMain	For internal use. (Inherited from Animation.)

Public Properties

	Name	Description
	IsRunning	Indicates whether the animation is currently running. (Inherited from Animation.)
	Item	Gets or sets the diagram item that should get animated. (Inherited from ItemAnimation.)
	Options	Gets or sets the object used to specify various property values for the animation. (Inherited from Animation.)
	Progress	Gets the progress of the current animation iteration. (Inherited from Animation.)

Public Methods

	Name	Description
	Dispose	IDisposable.Dispose implementation. (Inherited from Animation.)
	Start	Starts or resumes the animation. (Inherited from Animation.)
	Stop	Stops the animation. (Inherited from Animation.)

Protected Methods

	Name	Description
	FloatValue	Gets the current value from animation's range. (Inherited from Animation.)
	OnAnimationComplete	Raises the AnimationComplete event. (Inherited from Animation.)
	UpdateAnimation	Advance the animation. (Inherited from Animation.)

Public Events

	Name	Description
	AnimationComplete	Raised when an Animation completes. (Inherited from Animation.)

See Also

BounceAnimation Class
MindFusion.Diagramming.Animations Namespace

BounceAnimation Events

Animation.AnimationComplete Event

Raised when an Animation completes.

Namespace: MindFusion.Animations
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public event EventHandler AnimationComplete
```

Visual Basic Copy Code

```
Public Event AnimationComplete As EventHandler
```

See Also

Animation Members
Animation Class
MindFusion.Animations Namespace

FillColorAnimation Class

Animates the fill color of nodes or link arrowheads.

Syntax

JavaScript Copy Code

```
// class
FillColorAnimation.prototype = {}
```

Remarks

Call item's animate method to run an animation.

Example

JavaScript Copy Code

```
node.animate(
    new FillColorAnimation(
        {
            fromValue: new Drawing.Color("white"),
            toValue: new Drawing.Color("lightSkyBlue"),
            duration: 250,
```

```
            keepLastValue: true
        }));
```

Inheritance Hierarchy

MindFusion.Animations.Animation
  MindFusion.Diagramming.Animations.ItemAnimation
    MindFusion.Diagramming.Animations.FillColorAnimation

See Also

FillColorAnimation Members
MindFusion.Diagramming.Animations Namespace

FillColorAnimation Members

Animates the fill color of nodes or link arrowheads.

The following tables list the members exposed by the FillColorAnimation type.

Public Fields

	Name	Description
	AnimationsRunning	Track how many animations are running. (Inherited from Animation.)
	InvokeOnMain	For internal use. (Inherited from Animation.)

Public Properties

	Name	Description
	IsRunning	Indicates whether the animation is currently running. (Inherited from Animation.)
	Item	Gets or sets the diagram item that should get animated. (Inherited from ItemAnimation.)
	Options	Gets or sets the object used to specify various property values for the animation. (Inherited from Animation.)
	Progress	Gets the progress of the current animation iteration. (Inherited from Animation.)

Public Methods

	Name	Description
	Dispose	IDisposable.Dispose implementation. (Inherited from Animation.)
	Start	Starts or resumes the animation. (Inherited from Animation.)
	Stop	Stops the animation. (Inherited from Animation.)

Protected Methods

	Name	Description
	FloatValue	Gets the current value from animation's range. (Inherited from Animation.)
	OnAnimationComplete	Raises the AnimationComplete event. (Inherited from Animation.)
	UpdateAnimation	Advance the animation. (Inherited from Animation.)

Public Events

	Name	Description

	AnimationComplete	Raised when an Animation completes. (Inherited from Animation.)

See Also

FillColorAnimation Class
MindFusion.Diagramming.Animations Namespace

FillColorAnimation Events

ItemAnimation Class

Base class of diagram item animations.

Syntax

JavaScript Copy Code

```
// class
ItemAnimation.prototype = {}
```

Remarks

Call item's animate method to run an animation.

Inheritance Hierarchy

MindFusion.Animations.Animation
  MindFusion.Diagramming.Animations.ItemAnimation

See Also

ItemAnimation Members
MindFusion.Diagramming.Animations Namespace

ItemAnimation Members

Base class of diagram item animations.

The following tables list the members exposed by the ItemAnimation type.

Public Constructors

	Name	Description
	ItemAnimation	Initializes a new instance of the ItemAnimation class.

Public Fields

	Name	Description
	AnimationsRunning	Track how many animations are running. (Inherited from Animation.)
	InvokeOnMain	For internal use. (Inherited from Animation.)

Public Properties

	Name	Description
	IsRunning	Indicates whether the animation is currently running. (Inherited from Animation.)
	Item	Gets or sets the item to animate. (Inherited from Animation.)

	Options	Gets or sets the object used to specify various property values for the animation. (Inherited from Animation.)
	Progress	Gets the progress of the current animation iteration. (Inherited from Animation.)

Public Methods

	Name	Description
	Dispose	IDisposable.Dispose implementation. (Inherited from Animation.)
	Start	Starts or resumes the animation. (Inherited from Animation.)
	Stop	Stops the animation. (Inherited from Animation.)

Protected Methods

	Name	Description
	FloatValue	Gets the current value from animation's range. (Inherited from Animation.)
	OnAnimationComplete	Raises the AnimationComplete event. (Inherited from Animation.)
	UpdateAnimation	Advance the animation. (Inherited from Animation.)

Public Events

	Name	Description
	AnimationComplete	Raised when an Animation completes. (Inherited from Animation.)

See Also

ItemAnimation Class
MindFusion.Diagramming.Animations Namespace

ItemAnimation Constructor

Initializes a new instance of the ItemAnimation class.

Namespace: MindFusion.Diagramming.Animations
Assembly: ItemAnimation.js

Syntax

JavaScript Copy Code

```
function ItemAnimation ([options])
```

Parameters

options
Optional.
Object. An object used to specify various property values for the animation.

See Also

ItemAnimation Class
MindFusion.Diagramming.Animations Namespace

ItemAnimation Events

OpacityAnimation Class

Animates items' opacity for fade-in and fade-out effects.

Syntax

JavaScript Copy Code

```
// class
OpacityAnimation.prototype = {}
```

Remarks

Call item's animate method to run an animation.

Example

JavaScript Copy Code

```
node.animate(
    new OpacityAnimation(
        {
            fromValue: 1,
            toValue: 0.5,
            duration: 250,
            repeat: true,
            repeatCount: 4,
            reverse: true,
            animationType: Animations.AnimationType.BackBow
        }));
```

Inheritance Hierarchy

MindFusion.Animations.Animation
  MindFusion.Diagramming.Animations.ItemAnimation
    MindFusion.Diagramming.Animations.OpacityAnimation

See Also

OpacityAnimation Members
MindFusion.Diagramming.Animations Namespace

OpacityAnimation Members

Animates items' opacity for fade-in and fade-out effects.

The following tables list the members exposed by the OpacityAnimation type.

Public Fields

	Name	Description
	AnimationsRunning	Track how many animations are running. (Inherited from Animation.)
	InvokeOnMain	For internal use. (Inherited from Animation.)

Public Properties

	Name	Description
	IsRunning	Indicates whether the animation is currently running. (Inherited from Animation.)
	Item	Gets or sets the diagram item that should get animated. (Inherited from ItemAnimation.)

	Options	Gets or sets the object used to specify various property values for the animation. (Inherited from Animation.)
	Progress	Gets the progress of the current animation iteration. (Inherited from Animation.)

Public Methods

	Name	Description
	Dispose	IDisposable.Dispose implementation. (Inherited from Animation.)
	Start	Starts or resumes the animation. (Inherited from Animation.)
	Stop	Stops the animation. (Inherited from Animation.)

Protected Methods

	Name	Description
	FloatValue	Gets the current value from animation's range. (Inherited from Animation.)
	OnAnimationComplete	Raises the AnimationComplete event. (Inherited from Animation.)
	UpdateAnimation	Advance the animation. (Inherited from Animation.)

Public Events

	Name	Description
	AnimationComplete	Raised when an Animation completes. (Inherited from Animation.)

See Also

OpacityAnimation Class
MindFusion.Diagramming.Animations Namespace

OpacityAnimation Events

PathAnimation Class

Animates node positions along specified path.

Syntax

JavaScript Copy Code

```
// class
PathAnimation.prototype = {}
```

Remarks

Call item's animate method to run an animation.

Example

JavaScript Copy Code

```
node.animate(
    new PathAnimation(
        [
            new Point(30, 30),
            new Point(50, 30),
            new Point(75, 66),
```

```
            new Point(125, 88)
        ],
        {
            keepLastValue: true
        }));
```

Inheritance Hierarchy

MindFusion.Animations.Animation
  MindFusion.Diagramming.Animations.ItemAnimation
    MindFusion.Diagramming.Animations.PathAnimation

See Also

PathAnimation Members
MindFusion.Diagramming.Animations Namespace

PathAnimation Members

Animates node positions along specified path.

The following tables list the members exposed by the PathAnimation type.

Public Fields

	Name	Description
	AnimationsRunning	Track how many animations are running. (Inherited from Animation.)
	InvokeOnMain	For internal use. (Inherited from Animation.)

Public Properties

	Name	Description
	IsRunning	Indicates whether the animation is currently running. (Inherited from Animation.)
	Item	Gets or sets the diagram item that should get animated. (Inherited from ItemAnimation.)
	Options	Gets or sets the object used to specify various property values for the animation. (Inherited from Animation.)
	Progress	Gets the progress of the current animation iteration. (Inherited from Animation.)

Public Methods

	Name	Description
	Dispose	IDisposable.Dispose implementation. (Inherited from Animation.)
	Start	Starts or resumes the animation. (Inherited from Animation.)
	Stop	Stops the animation. (Inherited from Animation.)

Protected Methods

	Name	Description
	FloatValue	Gets the current value from animation's range. (Inherited from Animation.)
	OnAnimationComplete	Raises the AnimationComplete event. (Inherited from Animation.)
	UpdateAnimation	Advance the animation. (Inherited from Animation.)

Public Events

	Name	Description
	AnimationComplete	Raised when an Animation completes. (Inherited from Animation.)

See Also

PathAnimation Class
MindFusion.Diagramming.Animations Namespace

PathAnimation Events

PulseAnimation Class

Animates node's local scale factor.

Syntax

JavaScript Copy Code

```
// class
PulseAnimation.prototype = {}
```

Remarks

Call item's animate method to run an animation.

Example

JavaScript Copy Code

```
node.animate(
    new PulseAnimation(
        {
            fromValue: 1,
            toValue: 1.2,
            duration: 250,
            repeat: true,
            repeatCount: 4,
            reverse: true,
            animationType: Animations.AnimationType.BackBow
        }));
```

Inheritance Hierarchy

MindFusion.Animations.Animation
  MindFusion.Diagramming.Animations.ItemAnimation
    MindFusion.Diagramming.Animations.PulseAnimation

See Also

PulseAnimation Members
MindFusion.Diagramming.Animations Namespace

PulseAnimation Members

Animates node's local scale factor.

The following tables list the members exposed by the PulseAnimation type.

Public Fields

	Name	Description
	AnimationsRunning	Track how many animations are running. (Inherited from Animation.)
	InvokeOnMain	For internal use. (Inherited from Animation.)

Public Properties

	Name	Description
	IsRunning	Indicates whether the animation is currently running. (Inherited from Animation.)
	Item	Gets or sets the diagram item that should get animated. (Inherited from ItemAnimation.)
	Options	Gets or sets the object used to specify various property values for the animation. (Inherited from Animation.)
	Progress	Gets the progress of the current animation iteration. (Inherited from Animation.)

Public Methods

	Name	Description
	Dispose	IDisposable.Dispose implementation. (Inherited from Animation.)
	Start	Starts or resumes the animation. (Inherited from Animation.)
	Stop	Stops the animation. (Inherited from Animation.)

Protected Methods

	Name	Description
	FloatValue	Gets the current value from animation's range. (Inherited from Animation.)
	OnAnimationComplete	Raises the AnimationComplete event. (Inherited from Animation.)
	UpdateAnimation	Advance the animation. (Inherited from Animation.)

Public Events

	Name	Description
	AnimationComplete	Raised when an Animation completes. (Inherited from Animation.)

See Also

PulseAnimation Class
MindFusion.Diagramming.Animations Namespace

PulseAnimation Events

ScrollAnimation Class

Animates the scroll position of a diagram view.

Syntax

JavaScript Copy Code

```
// class
ScrollAnimation.prototype = {}
```

Remarks

Call view's animate method to run an animation.

Inheritance Hierarchy

MindFusion.Animations.Animation
  MindFusion.Diagramming.Animations.ViewAnimation
    MindFusion.Diagramming.Animations.ScrollAnimation

See Also

ScrollAnimation Members
MindFusion.Diagramming.Animations Namespace

ScrollAnimation Members

Animates the scroll position of a diagram view.

The following tables list the members exposed by the ScrollAnimation type.

Public Fields

	Name	Description
	AnimationsRunning	Track how many animations are running. (Inherited from Animation.)
	InvokeOnMain	For internal use. (Inherited from Animation.)

Public Properties

	Name	Description
	IsRunning	Indicates whether the animation is currently running. (Inherited from Animation.)
	Item	Gets or sets the item to animate. (Inherited from Animation.)
	Options	Gets or sets the object used to specify various property values for the animation. (Inherited from Animation.)
	Progress	Gets the progress of the current animation iteration. (Inherited from Animation.)

Public Methods

	Name	Description
	Dispose	IDisposable.Dispose implementation. (Inherited from Animation.)
	Start	Starts or resumes the animation. (Inherited from Animation.)
	Stop	Stops the animation. (Inherited from Animation.)

Protected Methods

	Name	Description
	FloatValue	Gets the current value from animation's range. (Inherited from Animation.)
	OnAnimationComplete	Raises the AnimationComplete event. (Inherited from Animation.)
	UpdateAnimation	Advance the animation. (Inherited from Animation.)

Public Events

	Name	Description
	AnimationComplete	Raised when an Animation completes. (Inherited from Animation.)

See Also

ScrollAnimation Class
MindFusion.Diagramming.Animations Namespace

ScrollAnimation Events

ShakeAnimation Class

Animates horizontal position of nodes.

Syntax

JavaScript Copy Code

```
// class
ShakeAnimation.prototype = {}
```

Remarks

Call item's animate method to run an animation.

Inheritance Hierarchy

MindFusion.Animations.Animation
  MindFusion.Diagramming.Animations.ItemAnimation
    MindFusion.Diagramming.Animations.ShakeAnimation

See Also

ShakeAnimation Members
MindFusion.Diagramming.Animations Namespace

ShakeAnimation Members

Animates horizontal position of nodes.

The following tables list the members exposed by the ShakeAnimation type.

Public Fields

	Name	Description
	AnimationsRunning	Track how many animations are running. (Inherited from Animation.)
	InvokeOnMain	For internal use. (Inherited from Animation.)

Public Properties

	Name	Description
	IsRunning	Indicates whether the animation is currently running. (Inherited from Animation.)
	Item	Gets or sets the diagram item that should get animated. (Inherited from ItemAnimation.)

	Options	Gets or sets the object used to specify various property values for the animation. (Inherited from Animation.)
	Progress	Gets the progress of the current animation iteration. (Inherited from Animation.)

Public Methods

	Name	Description
	Dispose	IDisposable.Dispose implementation. (Inherited from Animation.)
	Start	Starts or resumes the animation. (Inherited from Animation.)
	Stop	Stops the animation. (Inherited from Animation.)

Protected Methods

	Name	Description
	FloatValue	Gets the current value from animation's range. (Inherited from Animation.)
	OnAnimationComplete	Raises the AnimationComplete event. (Inherited from Animation.)
	UpdateAnimation	Advance the animation. (Inherited from Animation.)

Public Events

	Name	Description
	AnimationComplete	Raised when an Animation completes. (Inherited from Animation.)

See Also

ShakeAnimation Class
MindFusion.Diagramming.Animations Namespace

ShakeAnimation Events

StrokeColorAnimation Class

Animates the color of node borders or link segments.

Syntax

JavaScript Copy Code

```
// class
StrokeColorAnimation.prototype = {}
```

Remarks

Call item's animate method to run an animation.

Inheritance Hierarchy

MindFusion.Animations.Animation
  MindFusion.Diagramming.Animations.ItemAnimation
    MindFusion.Diagramming.Animations.StrokeColorAnimation

See Also

StrokeColorAnimation Members

Animates the color of node borders or link segments.

The following tables list the members exposed by the StrokeColorAnimation type.

Public Fields

	Name	Description
	AnimationsRunning	Track how many animations are running. (Inherited from Animation.)
	InvokeOnMain	For internal use. (Inherited from Animation.)

Public Properties

	Name	Description
	IsRunning	Indicates whether the animation is currently running. (Inherited from Animation.)
	Item	Gets or sets the diagram item that should get animated. (Inherited from ItemAnimation.)
	Options	Gets or sets the object used to specify various property values for the animation. (Inherited from Animation.)
	Progress	Gets the progress of the current animation iteration. (Inherited from Animation.)

Public Methods

	Name	Description
	Dispose	IDisposable.Dispose implementation. (Inherited from Animation.)
	Start	Starts or resumes the animation. (Inherited from Animation.)
	Stop	Stops the animation. (Inherited from Animation.)

Protected Methods

	Name	Description
	FloatValue	Gets the current value from animation's range. (Inherited from Animation.)
	OnAnimationComplete	Raises the AnimationComplete event. (Inherited from Animation.)
	UpdateAnimation	Advance the animation. (Inherited from Animation.)

Public Events

	Name	Description
	AnimationComplete	Raised when an Animation completes. (Inherited from Animation.)

See Also

StrokeColorAnimation Class
MindFusion.Diagramming.Animations Namespace

StrokeColorAnimation Events

ViewAnimation Class

Base class of diagram view animations.

Syntax

JavaScript Copy Code

```
// class
ViewAnimation.prototype = {}
```

Remarks

Call view's animate method to run an animation.

Inheritance Hierarchy

MindFusion.Animations.Animation
  MindFusion.Diagramming.Animations.ViewAnimation
    MindFusion.Diagramming.Animations.ScrollAnimation
    MindFusion.Diagramming.Animations.ZoomAnimation

See Also

ViewAnimation Members
MindFusion.Diagramming.Animations Namespace

ViewAnimation Members

Base class of diagram view animations.

The following tables list the members exposed by the ViewAnimation type.

Public Constructors

	Name	Description
	ViewAnimation	Initializes a new instance of the ItemAnimation class.

Public Fields

	Name	Description
	AnimationsRunning	Track how many animations are running. (Inherited from Animation.)
	InvokeOnMain	For internal use. (Inherited from Animation.)

Public Properties

	Name	Description
	IsRunning	Indicates whether the animation is currently running. (Inherited from Animation.)
	Item	Gets or sets the item to animate. (Inherited from Animation.)
	Options	Gets or sets the object used to specify various property values for the animation. (Inherited from Animation.)
	Progress	Gets the progress of the current animation iteration. (Inherited from Animation.)

Public Methods

	Name	Description
	Dispose	IDisposable.Dispose implementation. (Inherited from Animation.)
	Start	Starts or resumes the animation. (Inherited from Animation.)
	Stop	Stops the animation. (Inherited from Animation.)

Protected Methods

	Name	Description
	FloatValue	Gets the current value from animation's range. (Inherited from Animation.)
	OnAnimationComplete	Raises the AnimationComplete event. (Inherited from Animation.)
	UpdateAnimation	Advance the animation. (Inherited from Animation.)

Public Events

	Name	Description
	AnimationComplete	Raised when an Animation completes. (Inherited from Animation.)

See Also

ViewAnimation Class
MindFusion.Diagramming.Animations Namespace

ViewAnimation Constructor

Initializes a new instance of the ItemAnimation class.

Namespace: MindFusion.Diagramming.Animations
Assembly: ViewAnimation.js

Syntax

JavaScript — Copy Code

```
function ViewAnimation ([options])
```

Parameters

options
Optional.
Object. An object used to specify various property values for the animation.

See Also

ViewAnimation Class
MindFusion.Diagramming.Animations Namespace

ViewAnimation Events

ZoomAnimation Class

Animates the zoom level of a diagram view.

Syntax

JavaScript Copy Code

```
// class
ZoomAnimation.prototype = {}
```

Remarks

Call view's animate method to run an animation.

Inheritance Hierarchy

MindFusion.Animations.Animation
  MindFusion.Diagramming.Animations.ViewAnimation
    MindFusion.Diagramming.Animations.ZoomAnimation

See Also

ZoomAnimation Members
MindFusion.Diagramming.Animations Namespace

ZoomAnimation Members

Animates the zoom level of a diagram view.

The following tables list the members exposed by the ZoomAnimation type.

Public Fields

	Name	Description
	AnimationsRunning	Track how many animations are running. (Inherited from Animation.)
	InvokeOnMain	For internal use. (Inherited from Animation.)

Public Properties

	Name	Description
	IsRunning	Indicates whether the animation is currently running. (Inherited from Animation.)
	Item	Gets or sets the item to animate. (Inherited from Animation.)
	Options	Gets or sets the object used to specify various property values for the animation. (Inherited from Animation.)
	Progress	Gets the progress of the current animation iteration. (Inherited from Animation.)

Public Methods

	Name	Description
	Dispose	IDisposable.Dispose implementation. (Inherited from Animation.)
	Start	Starts or resumes the animation. (Inherited from Animation.)
	Stop	Stops the animation. (Inherited from Animation.)

Protected Methods

	Name	Description
	FloatValue	Gets the current value from animation's range. (Inherited from Animation.)
	OnAnimationComplete	Raises the AnimationComplete event. (Inherited from Animation.)
	UpdateAnimation	Advance the animation. (Inherited from Animation.)

Public Events

	Name	Description
	AnimationComplete	Raised when an Animation completes. (Inherited from Animation.)

See Also

ZoomAnimation Class
MindFusion.Diagramming.Animations Namespace

ZoomAnimation Events

MindFusion.Diagramming.Controls Namespace

Events Class

Defines all events raised by the controls in the MindFusion.Diagrammming.Controls namespace.

Syntax

JavaScript Copy Code

```
// class
Events.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Controls.Events

See Also

Events Members
MindFusion.Diagramming.Controls Namespace

Events Members

Defines all events raised by the controls in the MindFusion.Diagrammming.Controls namespace.

The following tables list the members exposed by the Events type.

Public Properties

	Name	Description
	pageAdded	Raised when the a page has been added to a TabbedDiagramView control.
	pageAdding	Raised before a page is added to a TabbedDiagramView control.
	pageChanged	Raised when the selected page in a TabbedDiagramView control has changed.
	pageChanging	Raised before the selected page in a TabbedDiagramView control is changed.
	pageRemoved	Raised when the a page has been removed from a TabbedDiagramView control.
	pageRemoving	Raised before a page is removed from a TabbedDiagramView control.

See Also

Events Class
MindFusion.Diagramming.Controls Namespace

Events Properties

Events.pageAdded Property

Raised when the a page has been added to a TabbedDiagramView control.

Namespace: MindFusion.Diagramming.Controls
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get pageAdded() {}
set pageAdded(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming.Controls Namespace

Events.pageAdding Property

Raised before a page is added to a TabbedDiagramView control.

Namespace: MindFusion.Diagramming.Controls
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get pageAdding() {}
set pageAdding(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming.Controls Namespace

Events.pageChanged Property

Raised when the selected page in a TabbedDiagramView control has changed.

Namespace: MindFusion.Diagramming.Controls
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get pageChanged() {}
set pageChanged(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming.Controls Namespace

Events.pageChanging Property

Raised before the selected page in a TabbedDiagramView control is changed.

Namespace: MindFusion.Diagramming.Controls
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get pageChanging() {}
set pageChanging(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming.Controls Namespace

Events.pageRemoved Property

Raised when the a page has been removed from a TabbedDiagramView control.

Namespace: MindFusion.Diagramming.Controls
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get pageRemoved() {}
set pageRemoved(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming.Controls Namespace

Events.pageRemoving Property

Raised before a page is removed from a TabbedDiagramView control.

Namespace: MindFusion.Diagramming.Controls
Assembly: Events.js

Syntax

JavaScript Copy Code

```
get pageRemoving() {}
set pageRemoving(value) {}
```

Property Value

String

A string containing the event name.

See Also

Events Members
Events Class
MindFusion.Diagramming.Controls Namespace

PageEventArgs Class

Contains the arguments passed to handlers of page-related events in a TabbedDiagramView.

Syntax

JavaScript Copy Code

```
// class
PageEventArgs.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Controls.CancelEventArgs
MindFusion.Diagramming.Controls.PageEventArgs

See Also

PageEventArgs Members
MindFusion.Diagramming.Controls Namespace

PageEventArgs Members

Contains the arguments passed to handlers of page-related events in a TabbedDiagramView.

The following tables list the members exposed by the PageEventArgs type.

Public Constructors

	Name	Description
	PageEventArgs	Initializes a new instance of the PageEventArgs class.

Public Properties

	Name	Description
	oldPage	Gets the page before the event.
	page	Gets the page for which the event is raised.

See Also

PageEventArgs Class
MindFusion.Diagramming.Controls Namespace

PageEventArgs Constructor

Initializes a new instance of the PageEventArgs class.

Namespace: MindFusion.Diagramming.Controls
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
function PageEventArgs (args)
```

Parameters

args

See Also

PageEventArgs Class
MindFusion.Diagramming.Controls Namespace

PageEventArgs Properties

PageEventArgs.oldPage Property

Gets the page before the event.

Namespace: MindFusion.Diagramming.Controls
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get oldPage() {}
set oldPage(value) {}
```

Property Value

DiagramPage

A DiagramPage instance.

See Also

PageEventArgs Members
PageEventArgs Class
MindFusion.Diagramming.Controls Namespace

PageEventArgs.page Property

Gets the page for which the event is raised.

Namespace: MindFusion.Diagramming.Controls
Assembly: EventArgs.js

Syntax

JavaScript Copy Code

```
get page() {}
set page(value) {}
```

Property Value

DiagramPage

A DiagramPage instance.

See Also

PageEventArgs Members
PageEventArgs Class
MindFusion.Diagramming.Controls Namespace

Palette Class

Represents a tool palette control that displays DiagramItem objects grouped into categories. Items from the palette can be added to a diagram by drag-and-drop.

Syntax

JavaScript Copy Code

```
// class
Palette.prototype = {}
```

Remarks

Content Security Policy

To allow for stricter CSP rules, the library no longer loads CSS scripts automatically. If you use **Palette**, you must explicitly link the *common-ui.css* file from distribution's *themes* folder.

The **Palette** is an **Accordion** subclass that displays ItemListView components as its child panels. Each **ItemListView** displays the items of a palette category.

Call the addCategory method to define a category and create its respective accordion pane. Call addItem to add an item to specified category.

Drag-and-drop operations create a copy of the dragged item, and set its size to newInstanceSize. The copy is created by calling item's clone method. You can override **clone** to copy property values of custom item subclasses.

Icons in the list are drawn by scaling **newInstanceSize** down to iconSize. E.g. if a list icon is half the size of the diagram item it creates, its graphic elements such as cells, labels, strokes would be rendered at half scale too (as opposed to rendering with same label / stroke sizes but in smaller bounding rectangle). The scaling code assumes measureUnit of the target diagram and the palette have matching values.

Configuring the control

Palette comes as s a part of the diagramming-controls npm package or the MindFusion.Diagramming.Controls namespace for UMD users.

For npm users simply import the diagramming-controls package:

JavaScript Copy Code

```
import * as Controls from '@mindfusion/diagramming-controls';
```

For UMD users, add references to files from distrbution's /umd folder using <script> tags. Note that the order of the scripts matters.

HTML Copy Code

```
<!-- for core Diagramming -->
<script src="Scripts/collections.js" type="text/javascript"></script>
```

```
<script src="Scripts/drawing.js" type="text/javascript"></script>
<script src="Scripts/controls.js" type="text/javascript"></script>
<script src="Scripts/graphs.js" type="text/javascript"></script>
<script src="Scripts/animations.js" type="text/javascript"></script>
<script src="Scripts/diagramming.js" type="text/javascript"></script>

<!-- for Palette -->
<script src="Scripts/common.js" type="text/javascript"></script>
<script src="Scripts/common-collections.js" type="text/javascript"></script>
<script src="Scripts/common-ui.js" type="text/javascript"></script>
<script src="Scripts/diagramming-controls.js" type="text/javascript"></script>
```

Creating the control instance and adding it to an HTML page can be done by calling the static create method:

JavaScript Copy Code

```
var palette = MindFusion.Diagramming.Controls.Palette
    .create(document.getElementById("palette"));
```

Default images for the collapse/expand buttons must be located in an ./icons folder. Alternatively, you can style the control with a predefined css theme, in which case the themes included in the distribution must be located in a ./themes folder and referenced in the head of the HTML document, e.g.

HTML / JavaScript Copy Code

```
<link rel="stylesheet" type="text/css" href="themes/common-ui.css" />
<link rel="stylesheet" type="text/css" href="themes/business.css" />
...
palette.theme = "business";
```

Inheritance Hierarchy

MindFusion.Common.UI.Accordion
  MindFusion.Diagramming.Controls.Palette

See Also

Palette Members
MindFusion.Diagramming.Controls Namespace

Palette Members

Represents a tool palette control that displays DiagramItem objects grouped into categories. Items from the palette can be added to a diagram by drag-and-drop.

The following tables list the members exposed by the Palette type.

Public Constructors

	Name	Description
	Palette	Initializes a new instance of the **Palette** class.

Public Properties

	Name	Description
	columns	Gets or sets the number of columns in palette's layout.
	iconSize	Gets or sets the size of items displayed in the palette.
	labelPosition	Gets or sets a value indicating the position of labels relative to item icons.
	measureUnit	Gets or sets the unit of measure used to specify item sizes.
	newInstanceSize	Gets or sets the default size of items created by drag-and-drop from this palette.

	padding	Gets or sets the padding space between item's slot bounds and its contents.
	textColor	Gets or sets the color used to paint item captions.

Public Methods

	Name	Description
	addCategory	Adds a new category to the palette.
	addItem	Adds an item to the palette.
S	create	Creates and initializes a new **Palette** from the specified element. This method is static and can be called without creating an instance of the class.
S	find	Returns the specified **Palette** object. This member is static and can be invoked without creating an instance of the class.
	removeCategory	Removes a category from the palette.
	removeItem	Removes an item from the palette.

See Also

Palette Class
MindFusion.Diagramming.Controls Namespace

Palette Constructor

Initializes a new instance of the Palette class.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript — Copy Code

```
function Palette (element)
```

Parameters

element
HTMLDivElement. Th DOM Element this Palette is associated with.

See Also

Palette Class
MindFusion.Diagramming.Controls Namespace

Palette Methods

Palette.addCategory Method

Adds a new category to the palette.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript — Copy Code

```
function addCategory (title, [image])
```

Parameters

title
String. The title text of the category.

image
Optional.
String. The URL of an image to be displayed in the category header.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace
addItem Method
removeCategory Method

Palette.addItem Method

Adds an item to the palette.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript Copy Code

```
function addItem (item, category, [caption])
```

Parameters

item
DiagramItem. The item to add.

category
String. The category of the item.

caption
Optional.
String. The caption to display for the item.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace
addCategory Method
removeItem Method

Palette.create Method

Creates and initializes a new Palette from the specified element. This method is static and can be called without creating an instance of the class.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript Copy Code

```
function create (element)
```

Parameters

element
HTMLDivElement. The DOM element that the Palette should be attached to.

Return Value

Palette. The newly created **Palette** object.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace

Palette.find Method

Returns the specified Palette object. This member is static and can be invoked without creating an instance of the class.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript Copy Code

```
function find (id)
```

Parameters

id
String. The id of the Palette's element.

Return Value

Palette. A **Palette** object with the specified id, if found, or **null** otherwise.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace

Palette.removeCategory Method

Removes a category from the palette.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript Copy Code

```
function removeCategory (title)
```

Parameters

title
String. The title of the category to remove.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace
addCategory Method

Palette.removeItem Method

Removes an item from the palette.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript Copy Code

```
function removeItem (item)
```

Parameters

item
DiagramItem. The item to remove.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace
addItem Method

Palette Properties

Palette.columns Property

Gets or sets the number of columns in palette's layout.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript Copy Code

```
get columns() {}
set columns(value) {}
```

Property Value

Number

An integer value specifying the number of columns. The default value is 1.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace

Palette.iconSize Property

Gets or sets the size of items displayed in the palette.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript Copy Code

```
get iconSize() {}
set iconSize(value) {}
```

Property Value

Size

An instance of the Size class specifying item size.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace
newInstanceSize Property
measureUnit Property

Palette.labelPosition Property

Gets or sets a value indicating the position of labels relative to item icons.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript Copy Code

```
get labelPosition() {}
set labelPosition(value) {}
```

Property Value

LabelPosition

A member of the LabelPosition enumeration. The default value is **Auto**.

Remarks

The default Auto value places labels to the right of item icons if using one-column layout, and below icons if showing multiple columns.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace

Palette.measureUnit Property

Gets or sets the unit of measure used to specify item sizes.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript Copy Code

```
get measureUnit() {}
set measureUnit(value) {}
```

Property Value

GraphicsUnit

A GraphicsUnit instance.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace
iconSize Property
newInstanceSize Property

Palette.newInstanceSize Property

Gets or sets the default size of items created by drag-and-drop from this palette.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript — Copy Code

```
get newInstanceSize() {}
set newInstanceSize(value) {}
```

Property Value

Size

An instance of the Size class specifying item size.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace
iconSize Property
measureUnit Property

Palette.padding Property

Gets or sets the padding space between item's slot bounds and its contents.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript — Copy Code

```
get padding() {}
set padding(value) {}
```

Property Value

Number

A number that specifies the padding space size.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace

Palette.textColor Property

Gets or sets the color used to paint item captions.

Namespace: MindFusion.Diagramming.Controls
Assembly: Palette.js

Syntax

JavaScript — Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner Class

Represents an interactive vector-drawing control that can be used to design node shapes.

Syntax

JavaScript Copy Code

```
// class
ShapeDesigner.prototype = {}
```

Remarks

To allow for stricter Content Security Policy rules, the library no longer loads CSS scripts automatically. If you use **ShapeDesigner**, you must explicitly link the *common-ui.css* file from distribution's *themes* folder.

Inheritance Hierarchy

MindFusion.Diagramming.Controls.ShapeDesigner

See Also

ShapeDesigner Members
MindFusion.Diagramming.Controls Namespace

ShapeDesigner Members

Represents an interactive vector-drawing control that can be used to design node shapes.

The following tables list the members exposed by the ShapeDesigner type.

Public Constructors

	Name	Description
	ShapeDesigner	Initializes a new instance of the ShapeDesigner class.

Public Properties

	Name	Description
	alignToGrid	Gets or sets a value indicating whether the diagram items should be aligned to a grid.
	anchorPointsLabel	Gets or sets a string specifying the label of graphic AnchorPoints category in shape designer toolbox.
	gridSizeX	Gets or sets the horizontal distance between adjacent grid points.
	gridSizeY	Gets or sets the vertical distance between adjacent grid points.
	primitivesLabel	Gets or sets a string specifying the label of graphic primitives category in shape designer toolbox.
	selectedElement	Gets or sets the selected element in the shape designer.
	showGrid	Gets or sets a value indicating whether the alignment grid is visible.

	showShapeProperties	Gets or sets a value indicating whether to display the property panel.
	templatesLabel	Gets or sets a string specifying the label of outline templates category in shape designer toolbox.
	theme	Gets or sets the theme of the control.
	themesPath	Gets or sets the path to the control's css themes.

Public Methods

	Name	Description
	copyToClipboard	Copies the currently selected element to the clipboard.
S	create	Creates and initializes a new ShapeDesigner from the specified element. This method is static and can be called without creating an instance of the class.
	exportShape	Creates a new Shape from the current content of the design canvas.
	importShape	Imports a Shape into the designer's drawing canvas.
	pasteFromClipboard	Pastes a previously copied element from the clipboard.
	redo	Executes again the most recently undone user action.
	undo	Undoes the most recent user action.

Public Events

	Name	Description
	shapeChanged	Raised when the shape has changed.

See Also

ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner Constructor

Initializes a new instance of the ShapeDesigner class.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
function ShapeDesigner (element)
```

Parameters

element
HTMLDivElement. The DOM Element this ShapeDesigner is associated with.

See Also

ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner Methods

ShapeDesigner.copyToClipboard Method

Copies the currently selected element to the clipboard.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
function copyToClipboard ()
```

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.create Method

Creates and initializes a new ShapeDesigner from the specified element. This method is static and can be called without creating an instance of the class.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
function create (element)
```

Parameters

element
HTMLDivElement. The DOM element that the ShapeDesigner should be attached to.

Return Value

ShapeDesigner. The newly created ShapeDesigner object.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.exportShape Method

Creates a new Shape from the current content of the design canvas.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
function exportShape (shapeId)
```

Parameters

shapeId
String. The shape's string identifier.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.importShape Method

Imports a Shape into the designer's drawing canvas.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
function importShape (shape)
```

Parameters

shape
Shape. A Shape instance.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.pasteFromClipboard Method

Pastes a previously copied element from the clipboard.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
function pasteFromClipboard ()
```

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.redo Method

Executes again the most recently undone user action.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
function redo ()
```

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.undo Method

Undoes the most recent user action.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript — Copy Code

```
function undo ()
```

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner Properties

ShapeDesigner.alignToGrid Property

Gets or sets a value indicating whether the diagram items should be aligned to a grid.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript — Copy Code

```
get alignToGrid() {}
set alignToGrid(value) {}
```

Property Value

Boolean

true if the diagram items should be aligned to a grid, or false otherwise.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.anchorPointsLabel Property

Gets or sets a string specifying the label of graphic AnchorPoints category in shape designer toolbox.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript — Copy Code

```
get anchorPointsLabel() {}
set anchorPointsLabel(value) {}
```

Property Value

String

The label of graphic AnchorPoints category.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.gridSizeX Property

Gets or sets the horizontal distance between adjacent grid points.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
get gridSizeX() {}
set gridSizeX(value) {}
```

Property Value

Number

The horizontal distance between adjacent grid points.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.gridSizeY Property

Gets or sets the vertical distance between adjacent grid points.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
get gridSizeY() {}
set gridSizeY(value) {}
```

Property Value

Number

The vertical distance between adjacent grid points.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.primitivesLabel Property

Gets or sets a string specifying the label of graphic primitives category in shape designer toolbox.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
get primitivesLabel() {}
set primitivesLabel(value) {}
```

Property Value

String

The label of graphic primitives category.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.selectedElement Property

Gets or sets the selected element in the shape designer.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
get selectedElement() {}
set selectedElement(value) {}
```

Property Value

InteractiveElement

An InteractiveElement instance.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.showGrid Property

Gets or sets a value indicating whether the alignment grid is visible.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
get showGrid() {}
set showGrid(value) {}
```

Property Value

Boolean

true if the alignment grid is visible, or false otherwise.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.showShapeProperties Property

Gets or sets a value indicating whether to display the property panel.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
get showShapeProperties() {}
set showShapeProperties(value) {}
```

Property Value

Boolean

true to display the property panel, or false otherwise. The default value is true.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.templatesLabel Property

Gets or sets a string specifying the label of outline templates category in shape designer toolbox.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
get templatesLabel() {}
set templatesLabel(value) {}
```

Property Value

String

The label of outline templates category.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.theme Property

Gets or sets the theme of the control.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
get theme() {}
set theme(value) {}
```

Property Value

Theme

An instance of the Theme class.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner.themesPath Property

Gets or sets the path to the control's css themes.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

JavaScript Copy Code

```
get themesPath() {}
set themesPath(value) {}
```

Property Value

String

A string specifying the path.

Remarks

Only applies to web components.

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

ShapeDesigner Events

ShapeDesigner.shapeChanged Event

Raised when the shape has changed.

Namespace: MindFusion.Diagramming.Controls
Assembly: ShapeDesigner.js

Syntax

See Also

ShapeDesigner Members
ShapeDesigner Class
MindFusion.Diagramming.Controls Namespace

TabAlignment Enumeration

Specifies the alignment of a tab strip relative to its container control.

Namespace: MindFusion.Diagramming.Controls
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
TabAlignment = {}
```

See Also

MindFusion.Diagramming.Controls Namespace

TabbedDiagramView Class

Represents a view that displays DiagramDocument objects.

Syntax

JavaScript — Copy Code

```
// class
TabbedDiagramView.prototype = {}
```

Remarks

Content Security Policy

To allow for stricter CSP rules, the library no longer loads CSS scripts automatically. If you use **TabbedDiagramView**, you must explicitly link the *common-ui.css* file from distribution's *themes* folder.

Individual diagram pages can be activated through the tabs located at the top or at the bottom of the control. The appearance and behavior of the tab strip can be customized using various properties. The location and visibility of the strip is specified through the tabAlignment property. Its height can be changed via the tabsSize property. showAddButton and showNavigationButtons determine the visibility of the buttons for interactive tab creation and tab navigation respectively. showTabCloseButtons specifies whether the tabs can be closed interactively, by clicking on the small 'x' button displayed in each tab. allowReorder enables interactive tab rearrangement by mouse dragging.

Configuring the control

TabbedDiagramView comes as s a part of the diagramming-controls npm package or the MindFusion.Diagramming.Controls namespace for UMD users.

For npm users simply import the diagramming-controls package:

JavaScript — Copy Code

```
import * as Controls from '@mindfusion/diagramming-controls';
```

For UMD users, add references to files from distrbution's /umd folder using <script> tags. Note that the order of the scripts matters.

HTML — Copy Code

```
<!-- for core Diagramming -->
<script src="Scripts/collections.js" type="text/javascript"></script>
<script src="Scripts/drawing.js" type="text/javascript"></script>
<script src="Scripts/controls.js" type="text/javascript"></script>
<script src="Scripts/graphs.js" type="text/javascript"></script>
<script src="Scripts/animations.js" type="text/javascript"></script>
<script src="Scripts/diagramming.js" type="text/javascript"></script>

<!-- for TabbedDiagramView -->
<script src="Scripts/common.js" type="text/javascript"></script>
<script src="Scripts/common-collections.js" type="text/javascript"></script>
```

```
<script src="Scripts/common-ui.js" type="text/javascript"></script>
<script src="Scripts/diagramming-controls.js" type="text/javascript"></script>
```

Creating the control instance and adding it to an HTML page is the same as for the DiagramView control. You can pass a DiagramDocument instance to the create function, or omit that parameter to let the control create a new document with a single DiagramPage.

JavaScript Copy Code

```
var doc = new MindFusion.Diagramming.DiagramDocument();
var page1 = new MindFusion.Diagramming.DiagramPage("diagram1");
var node = page1.factory.createShapeNode(10, 10, 20, 20);
doc.addPage(page1);
var diagramView = MindFusion.Diagramming.Controls.TabbedDiagramView.create(
    document.getElementById("diagram"), doc);
```

or simply

JavaScript Copy Code

```
var diagramView = MindFusion.Diagramming.Controls.TabbedDiagramView.create(
    document.getElementById("diagram"));
```

You can specify a title, iconUrl and titleColor for diagram pages by setting the corresponding properties.

Default images for the navigation buttons must be located in an ./icons folder. Alternatively, you can style the control with a predefined css theme, in which case the themes included in the distribution must be located in a ./themes folder and referenced in the head of the HTML document, e.g.

HTML / JavaScript Copy Code

```
<link rel="stylesheet" type="text/css" href="themes/common-ui.css" />
<link rel="stylesheet" type="text/css" href="themes/business.css" />
...
diagramView.theme = MindFusion.Diagramming.Theme.fromId("business");
```

Inheritance Hierarchy

MindFusion.Diagramming.DiagramView
MindFusion.Diagramming.Controls.TabbedDiagramView

See Also

TabbedDiagramView Members
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView Members

Represents a view that displays DiagramDocument objects.

The following tables list the members exposed by the TabbedDiagramView type.

Public Constructors

	Name	Description
	TabbedDiagramView	Initializes a new instance of the TabbedDiagramView class.

Public Properties

	Name	Description
	allowRenamePages	Gets or sets a value indicating whether DiagramPage titles can be renamed interactively.
	allowReorder	Gets or sets a value indicating whether tabs can be rearranged interactively.
	document	Gets or sets the diagram document displayed by this view.

	selectedIndex	Gets or sets the index of the currently selected diagram page.
	selectedPage	Gets or sets the currently selected tab page.
	showAddButton	Gets or sets a value indicating whether to display the 'Add Tab' button.
	showNavigationButtons	Gets or sets a value indicating whether to show tab navigation buttons.
	showTabCloseButtons	Gets or sets a value indicating whether to display a close button for each tab.
	tabAlignment	Gets or sets the tab alignment.
	tabsSize	Gets or sets the height of the tab strip.
	theme	Gets or sets the theme of the control.
	themesPath	Gets or sets the path to the control's css themes.

Public Methods

	Name	Description
	create	Creates and initializes a new TabbedDiagramView from the specified element. This method is static and can be called without creating an instance of the class.
	dispose	DiagramView.dispose override.
	fromJson	Loads the diagram from a JSON string.
	initialize	DiagramView.initialize override.
	loadFromXml	Loads the diagram from an XML file.
	saveToXml	Saves the diagram to an XML file.
	setTheme	Sets the current theme of the control and its pages.
	toJson	Saves the diagram to a JSON string.

See Also

TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView Constructor

Initializes a new instance of the TabbedDiagramView class.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
function TabbedDiagramView (element, [document])
```

Parameters

element
HTMLCanvasElement. The Canvas DOM Element this TabbedDiagramView is associated with.

document
Optional.
DiagramDocument. The DiagramDocument instance this control is associated with.

See Also

TabbedDiagramView Methods

TabbedDiagramView.create Method

Creates and initializes a new TabbedDiagramView from the specified element. This method is static and can be called without creating an instance of the class.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
function create (element, [document])
```

Parameters

element
HTMLCanvasElement. The DOM element that the DiagramView should be attached to.

document
Optional.
DiagramDocument. The DiagramDocument instance this control is associated with.

Return Value

TabbedDiagramView. The newly created TabbedDiagramView object.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.dispose Method

DiagramView.dispose override.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
function dispose ()
```

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.fromJson Method

Loads the diagram from a JSON string.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
function fromJson ()
```

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.initialize Method

DiagramView.initialize override.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
function initialize ()
```

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.loadFromXml Method

Loads the diagram from an XML file.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
function loadFromXml (fileUrl, [onLoad, [onError]])
```

Parameters

fileUrl
String. The URL of an XML file where the data should be read from.

onLoad
Optional.
function. A callback that should be invoked if the file is loaded successfully.

onError
Optional.
function. A callback that should be invoked if the file could not be downloaded.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.saveToXml Method

Saves the diagram to an XML file.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
function saveToXml (url)
```

Parameters

url
String. A URL specifying where the diagram's XML should be posted to.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.setTheme Method

Sets the current theme of the control and its pages.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
function setTheme (theme, [setToPages])
```

Parameters

theme
Theme. The theme.

setToPages
Optional.
Boolean. true to set pages' themes, otherwise false.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.toJson Method

Saves the diagram to a JSON string.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
function toJson ()
```

Return Value

String. A string containing the diagram's JSON representation.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView Properties

TabbedDiagramView.allowRenamePages Property

Gets or sets a value indicating whether DiagramPage titles can be renamed interactively.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get allowRenamePages() {}
set allowRenamePages(value) {}
```

Property Value

Boolean

true if tabs can be renamed interactively, or **false** otherwise. The default value is **false**.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.allowReorder Property

Gets or sets a value indicating whether tabs can be rearranged interactively.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get allowReorder() {}
set allowReorder(value) {}
```

Property Value

Boolean

true if tabs can be rearranged interactively, or false otherwise. The default value is false.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.document Property

Gets or sets the diagram document displayed by this view.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get document() {}
set document(value) {}
```

Property Value

DiagramDocument

The DiagramDocument instance displayed by this view.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.selectedIndex Property

Gets or sets the index of the currently selected diagram page.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get selectedIndex() {}
set selectedIndex(value) {}
```

Property Value

Number

The index of the selected diagram page.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.selectedPage Property

Gets or sets the currently selected tab page.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get selectedPage() {}
set selectedPage(value) {}
```

Property Value

DiagramPage

The selected diagram page.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.showAddButton Property

Gets or sets a value indicating whether to display the 'Add Tab' button.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get showAddButton() {}
set showAddButton(value) {}
```

Property Value

Boolean

true if the 'Add Tab' button is visible, or false otherwise. The default value is false.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.showNavigationButtons Property

Gets or sets a value indicating whether to show tab navigation buttons.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get showNavigationButtons() {}
set showNavigationButtons(value) {}
```

Property Value

Boolean

true if the tab navigation buttons are visible, or false otherwise. The default value is false.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.showTabCloseButtons Property

Gets or sets a value indicating whether to display a close button for each tab.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get showTabCloseButtons() {}
set showTabCloseButtons(value) {}
```

Property Value

Boolean

true if the tabs close buttons are visible, or false otherwise. The default value is false.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.tabAlignment Property

Gets or sets the tab alignment.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get tabAlignment() {}
set tabAlignment(value) {}
```

Property Value

TabAlignment

A member of the TabAlignment enumeration.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.tabsSize Property

Gets or sets the height of the tab strip.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get tabsSize() {}
set tabsSize(value) {}
```

Property Value

Number

The height.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.theme Property

Gets or sets the theme of the control.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get theme() {}
set theme(value) {}
```

Property Value

Theme

An instance of the Theme class.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

TabbedDiagramView.themesPath Property

Gets or sets the path to the control's css themes.

Namespace: MindFusion.Diagramming.Controls
Assembly: TabbedDiagramView.js

Syntax

JavaScript Copy Code

```
get themesPath() {}
set themesPath(value) {}
```

Property Value

String

A string specifying the path.

Remarks

Only applies to web components.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Controls Namespace

WebComponents Class

Implements registration of web components.

Syntax

JavaScript Copy Code

```
// class
WebComponents.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Controls.WebComponents

See Also

WebComponents Members
MindFusion.Diagramming.Controls Namespace

WebComponents Members

Implements registration of web components.

The following tables list the members exposed by the WebComponents type.

Public Methods

	Name	Description
	register	Registers the controls in the package as web components.

See Also

WebComponents Class
MindFusion.Diagramming.Controls Namespace

WebComponents Methods

WebComponents.register Method

Registers the controls in the package as web components.

Namespace: MindFusion.Diagramming.Controls
Assembly: WebComponents.js

Syntax

JavaScript Copy Code

```
function register ()
```

Remarks

Call this method to register the TabbedDiagramView class as a web component. After registration, the browser instantiates **TabbedDiagramView** objects for <mindfusion-tabbeddiagramview> HTML tags. You can get reference to objects created by the browser by calling the find method.

See Also

WebComponents Members
WebComponents Class
MindFusion.Diagramming.Controls Namespace
create Method

MindFusion.Diagramming.Lanes Namespace

Cell Class

Represents a single cell in a lane grid.

Syntax

JavaScript Copy Code

```
// class
Cell.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Lanes.Cell

See Also

Cell Members
MindFusion.Diagramming.Lanes Namespace

Cell Members

Represents a single cell in a lane grid.

The following tables list the members exposed by the Cell type.

Public Constructors

	Name	Description
	Cell	Initializes a new instance of the **Cell** class. For internal use only.

Public Properties

	Name	Description
	style	Gets or sets the style of the cell.
	text	Gets or sets the text of the cell.

Public Methods

	Name	Description
	loadFromXml	Deserializes the cell from the specified XML element
	saveToXml	Serializes the cell matrix to the specified XML element

See Also

Cell Class
MindFusion.Diagramming.Lanes Namespace

Cell Constructor

Initializes a new instance of the Cell class. For internal use only.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
function Cell ()
```

See Also

Cell Methods

Cell.loadFromXml Method

Deserializes the cell from the specified XML element

Namespace: MindFusion.Diagramming.Lanes
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

Cell Members
Cell Class
MindFusion.Diagramming.Lanes Namespace

Cell.saveToXml Method

Serializes the cell matrix to the specified XML element

Namespace: MindFusion.Diagramming.Lanes
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

Cell Members
Cell Class
MindFusion.Diagramming.Lanes Namespace

Cell Properties

Cell.style Property

Gets or sets the style of the cell.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get style() {}
set style(value) {}
```

Property Value

Lanes.Style

A Style object that specifies the style of this cell.

See Also

Cell Members
Cell Class
MindFusion.Diagramming.Lanes Namespace

Cell.text Property

Gets or sets the text of the cell.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Cell.js

Syntax

JavaScript Copy Code

```
get text() {}
set text(value) {}
```

Property Value

String

A string specifying the text of the cell.

See Also

Cell Members
Cell Class
MindFusion.Diagramming.Lanes Namespace

Grid Class

Represents a lane grid.

Syntax

JavaScript Copy Code

```
// class
Grid.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Lanes.Grid

See Also

Grid Members
MindFusion.Diagramming.Lanes Namespace

Grid Members

Represents a lane grid.

The following tables list the members exposed by the Grid type.

Public Constructors

	Name	Description
	Grid	Initializes a new instance of the **Grid** class. For internal use only.

Public Properties

	Name	Description
	allowInplaceEdit	Gets or sets a value indicating whether grid cells' text can be edited interactively.
	allowResizeHeaders	Gets or sets a value indicating whether users can resize the headers of the grid interactively.
	columnCount	Gets or sets the number of columns in the grid.
	columnHeaderBounds	Returns the bounding rectangle of all column headers.
	columnHeaders	Gets the collection of topmost column headers.
	columnHeadersHeights	Gets or sets an array of numbers specifying the heights of individual column levels.
	headersOnTop	Gets or sets a value indicating whether lane headers appear at the top of the z-order above all diagram items.
	hookHeaders	Gets or sets a value indicating whether headers will be anchored to their corresponding side.
	leftMargin	Gets or sets the offset of the lane grid from left side of diagram's bounds.
	minHeaderSize	Gets or sets the minimum size of a header.
	rowCount	Gets or sets the number of rows in the grid.
	rowHeaderBounds	Returns the bounding rectangle of all row headers.
	rowHeaders	Gets the collection of topmost row headers.
	rowHeadersWidths	Gets or sets an array of numbers specifying the widths of individual row levels.
	style	Gets the default style of the grid.
	topLeftAreaStyle	Gets or sets the style of the area above all row headers and to the left of all column headers.
	topLeftAreaText	Gets or sets the text displayed in the top left area.
	topMargin	Gets or sets the offset of the lane grid from top side of diagram's bounds.
	totalColumnHeight	Gets the total height of all column headers.
	totalRowWidth	Gets the total width of all row headers.

Public Methods

	Name	Description
	addColumnHeader	Adds the specified header to the grid as a new column header.
	addRowHeader	Adds the specified header to the grid as a new row header.
	findColumn	Finds the first column whose header caption matches the specified string.
	findRow	Finds the first row whose header caption matches the specified string.
	get	Gets the cell or the cluster of cells corresponding to the specified row and column.
	getCellBounds	Returns the bounding rectangle of the specified cell.
	getCellColumn	Returns the column index of the specified cell.
	getCellFromPoint	Gets the cell located at the specified point.
	getCellRow	Returns the row index of the specified cell.
	getColumn	Returns the column header with the specified index from the bottommost column header level.
	getColumnDepth	Gets the number of levels of the column headers.
	getColumnIndex	Returns the 0-based index of the specified column. Only works for bottommost headers. If the specified column is a group header, returns -1.
	getHeaderBounds	Returns the bounding rectangle of the specified header.
	getHeaderFromPoint	Gets the Header that is located at the specified point.
	getRow	Returns the row header with the specified index from the bottommost row header level.
	getRowDepth	Gets the number of levels of the row headers.
	getRowIndex	Returns the 0-based index of the specified row. Only works for bottommost headers. If the specified row is a group header, returns -1;
	loadFromXml	Deserializes the lane grid from the specified XML element.
	removeColumnHeader	Removes the specified column header and all of its associated cells from the grid.
	removeRowHeader	Removes the specified row header and all of its associated cells from the grid.
	saveToXml	Serializes the lane grid to the specified XML element.

See Also

Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid Constructor

Initializes a new instance of the Grid class. For internal use only.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function Grid ()
```

See Also

Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid Methods

Grid.addColumnHeader Method

Adds the specified header to the grid as a new column header.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript — Copy Code

```
function addColumnHeader (header, [parent, [i]])
```

Parameters

header
Lanes.Header. The Header object to add.

parent
Optional.
Lanes.Header. The parent header to add to, or **null** to add the new header as a root column.

i
Optional.
Number. The index at which the header was added in the collection.

Example

The following example adds a new header at the end of the column collection of an existing lane grid.

JavaScript — Copy Code

```
grid.addColumnHeader(new Header());
```

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.addRowHeader Method

Adds the specified header to the grid as a new row header.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript — Copy Code

```
function addRowHeader (header, [parent, [i]])
```

Parameters

header
Lanes.Header. The Header object to add.

parent
Optional.
Lanes.Header. The parent header to add to, or **null** to add the new header as a root row.

i

Optional.
Number. The index at which the header was added in the collection.

Example

The following example adds a new header at the end of the row collection of an existing lane grid.

JavaScript Copy Code

```
d.addRowHeader(new Header());
```

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.findColumn Method

Finds the first column whose header caption matches the specified string.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function findColumn (title)
```

Parameters

title
String. A string containing the header title to look for.

Return Value

Returns the first column Header that has the specified caption.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.findRow Method

Finds the first row whose header caption matches the specified string.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function findRow (title)
```

Parameters

title
String. A string containing the header title to look for.

Return Value

Returns the first row Header that has the specified caption.

See Also

Grid.get Method

Gets the cell or the cluster of cells corresponding to the specified row and column.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function get (column, row)
```

Parameters

column
Number | Lanes.Header. A Header object or a cell index.

row
Number | Lanes.Header. A **Header** object or a cell index.

Return Value

Lanes.Cell. A Cell object representing the cell or cluster of cells with the specified coordinates.

Remarks

This method accepts either Header objects or numbers as arguments, but not a mix of both. If more than one cell corresponds to specified **Header** object, this method returns an object identifying all corresponding cells. Modifying the style of the returned object will affect all cells in the cluster. Calling the method without arguments, or with **null** arguments, will return an object identifying all cells in the grid.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.getCellBounds Method

Returns the bounding rectangle of the specified cell.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function getCellBounds (cell)
```

Parameters

cell
Lanes.Cell. A Cell object representing a cell or a group of cells in the grid.

Return Value

Rect. A Rect instance representing the rectangle that bounds the specified cell or group of cells.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.getCellColumn Method

Returns the column index of the specified cell.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript — Copy Code

```
function getCellColumn (cell)
```

Parameters

cell
Lanes.Cell. A Cell object representing a single cell in the grid.

Return Value

Number. The index of the column that contains cell; -1 if cell is not found or refers to a group of cells.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.getCellFromPoint Method

Gets the cell located at the specified point.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript — Copy Code

```
function getCellFromPoint (point, cellBounds)
```

Parameters

point
A Point instance specifying a point within the grid.

cellBounds
An object that receives the coordinates of the found cell in its **cellBounds** field.

Return Value

A Cell object identifying the found cell, or **null** if there is no cell located at the specified point.

Example

The following example illustrates how to find the cell at a specific position as well as its bounding rectangle:

JavaScript — Copy Code

```
var boundsResult = {};
var cell = grid.getCellFromPoint(new Point(50, 50), boundsResult);
if (cell)
{
    var cellBounds = boundsResult.cellBounds;
}
```

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.getCellRow Method

Returns the row index of the specified cell.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript — Copy Code

```
function getCellRow (cell)
```

Parameters

cell
Lanes.Cell. A Cell object representing a single cell in the grid.

Return Value

Number. The index of the row that contains cell; -1 if cell is not found or refers to a group of cells.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.getColumn Method

Returns the column header with the specified index from the bottommost column header level.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript — Copy Code

```
function getColumn (index)
```

Parameters

index
Number. A number specifying the column index.

Return Value

A Header object representing the column at the specified index.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.getColumnDepth Method

Gets the number of levels of the column headers.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript — Copy Code

```
function getColumnDepth ()
```

Return Value

Number. A number representing the column depth.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.getColumnIndex Method

Returns the 0-based index of the specified column. Only works for bottommost headers. If the specified column is a group header, returns -1.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function getColumnIndex (column)
```

Parameters

column

Lanes.Header. A Header instance that represents a grid column.

Return Value

Number. The index of column, or -1 if the specified header represents a row or a group of child columns.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.getHeaderBounds Method

Returns the bounding rectangle of the specified header.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function getHeaderBounds (header, includeSubHeaders)
```

Parameters

header

Lanes.Header. The Header object whose coordinates to return.

includeSubHeaders

Boolean. **true** to include the coordinates of the child headers in the returned rectangle, or **false** otherwise.

Return Value

Rect. A Rect that bounds the specified header, and optionally its child headers.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.getHeaderFromPoint Method

Gets the Header that is located at the specified point.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript — Copy Code

```
function getHeaderFromPoint (point, headerBounds, scrollPosition)
```

Parameters

point
A Point instance specifying a point within the grid.

headerBounds
An object that receives the coordinates of the found header in its **headerBounds** field.

scrollPosition
Point. When the headers are hooked this parameter should provide information about the current scroll position in the view.

Return Value

The found Header object, or **null** if there is no header located at the specified point.

Example

The following example illustrates how to find the header at a specific position as well as its bounding rectangle:

JavaScript — Copy Code

```
var boundsResult = {};
var header = grid.getHeaderFromPoint(new Point(20, 10), boundsResult);
if (header)
{
    var headerBounds = boundsResult.headerBounds;
}
```

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.getRow Method

Returns the row header with the specified index from the bottommost row header level.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript — Copy Code

```
function getRow (index)
```

Parameters

index
Number. A number specifying the row index.

Return Value

A Header object representing the row at the specified index.

See Also

Grid Members
Grid Class

Grid.getRowDepth Method

Gets the number of levels of the row headers.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function getRowDepth ()
```

Return Value

Number. A number representing the row depth.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.getRowIndex Method

Returns the 0-based index of the specified row. Only works for bottommost headers. If the specified row is a group header, returns -1;

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function getRowIndex (row)
```

Parameters

row
Lanes.Header. A Header instance that represents a grid row.

Return Value

Number. The index of row, or -1 if the specified header represents a column or a group of child rows.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.loadFromXml Method

Deserializes the lane grid from the specified XML element.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the deserialization process and some helper deserialization methods.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.removeColumnHeader Method

Removes the specified column header and all of its associated cells from the grid.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function removeColumnHeader (header)
```

Parameters

header
Lanes.Header. The Header object to remove.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.removeRowHeader Method

Removes the specified row header and all of its associated cells from the grid.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function removeRowHeader (header)
```

Parameters

header
Lanes.Header. The Header object to remove.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.saveToXml Method

Serializes the lane grid to the specified XML element.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement

Element. An XML DOM element that will contain the item's data.

context

XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid Properties

Grid.allowInplaceEdit Property

Gets or sets a value indicating whether grid cells' text can be edited interactively.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get allowInplaceEdit() {}
set allowInplaceEdit(value) {}
```

Property Value

Boolean

true if grid cells' text can be in-place edited, or **false** otherwise.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.allowResizeHeaders Property

Gets or sets a value indicating whether users can resize the headers of the grid interactively.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get allowResizeHeaders() {}
set allowResizeHeaders(value) {}
```

Property Value

Boolean

true if grid headers can resized interactively, or **false** otherwise.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.columnCount Property

Gets or sets the number of columns in the grid.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get columnCount() {}
set columnCount(value) {}
```

Property Value

Number

The number of columns.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.columnHeaderBounds Property

Returns the bounding rectangle of all column headers.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get columnHeaderBounds() {}
```

Property Value

Rect

A Rect that bounds all column headers.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.columnHeaders Property

Gets the collection of topmost column headers.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get columnHeaders() {}
```

Property Value

Array

An array containing the topmost column headers.

Remarks

The returned array should not be modified directly. To add and remove column headers to the grid, use the addColumnHeader and removeColumnHeader methods respectively.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.columnHeadersHeights Property

Gets or sets an array of numbers specifying the heights of individual column levels.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get columnHeadersHeights() {}
set columnHeadersHeights(value) {}
```

Property Value

Array

An array of numbers representing the heights of the various header levels.

Remarks

If a value is non-positive or does not exist because the array has less arguments than the number of columns, then the height is calculated as the maximum height of the columns in that level.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.headersOnTop Property

Gets or sets a value indicating whether lane headers appear at the top of the z-order above all diagram items.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get headersOnTop() {}
set headersOnTop(value) {}
```

Property Value

Boolean

true to display headers on top of items, or **false** otherwise.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.hookHeaders Property

Gets or sets a value indicating whether headers will be anchored to their corresponding side.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get hookHeaders() {}
set hookHeaders(value) {}
```

Property Value

Boolean

true to anchor the headers, or **false** otherwise.

Remarks

If enabled, the column header stays anchored to the top side and the row header anchored to the left side of diagram's viewport, regardless of scroll position.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.leftMargin Property

Gets or sets the offset of the lane grid from left side of diagram's bounds.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get leftMargin() {}
set leftMargin(value) {}
```

Property Value

Number

A number representing the left margin.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.minHeaderSize Property

Gets or sets the minimum size of a header.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get minHeaderSize() {}
set minHeaderSize(value) {}
```

Property Value

Number

A number representing the minimum size for headers in the grid.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.rowCount Property

Gets or sets the number of rows in the grid.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get rowCount() {}
set rowCount(value) {}
```

Property Value

Number

The number of rows.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.rowHeaderBounds Property

Returns the bounding rectangle of all row headers.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get rowHeaderBounds() {}
```

Property Value

Rect

A Rect that bounds all row headers.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.rowHeaders Property

Gets the collection of topmost row headers.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get rowHeaders() {}
set rowHeaders(value) {}
```

Property Value

Array

An array containing the topmost row headers.

Remarks

The returned array should not be modified directly. To add and remove row headers to the grid, use the addRowHeader and removeRowHeader methods respectively.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.rowHeadersWidths Property

Gets or sets an array of numbers specifying the widths of individual row levels.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get rowHeadersWidths() {}
set rowHeadersWidths(value) {}
```

Property Value

Array

An array of numbers representing the widths of the various header levels.

Remarks

If a value is non-positive or does not exist because the array has less arguments than the number of rows, then the width is calculated as the maximum width of the rows in that level.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.style Property

Gets the default style of the grid.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get style() {}
```

Property Value

Lanes.Style

A Style object representing the style of all grid elements, which do not have an explicitly assigned style.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.topLeftAreaStyle Property

Gets or sets the style of the area above all row headers and to the left of all column headers.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get topLeftAreaStyle() {}
set topLeftAreaStyle(value) {}
```

Property Value

Lanes.Style

A Style object that specifies the appearance of the top-left area.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.topLeftAreaText Property

Gets or sets the text displayed in the top left area.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get topLeftAreaText() {}
set topLeftAreaText(value) {}
```

Property Value

String

A string containing the text.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.topMargin Property

Gets or sets the offset of the lane grid from top side of diagram's bounds.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get topMargin() {}
set topMargin(value) {}
```

Property Value

Number

A number representing the top margin.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.totalColumnHeight Property

Gets the total height of all column headers.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get totalColumnHeight() {}
```

Property Value

Number

A number representing the height.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Grid.totalRowWidth Property

Gets the total width of all row headers.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Grid.js

Syntax

JavaScript Copy Code

```
get totalRowWidth() {}
```

Property Value

Number

A number representing the width.

See Also

Grid Members
Grid Class
MindFusion.Diagramming.Lanes Namespace

Header Class

Represents a header in a lane grid.

Syntax

JavaScript Copy Code

```
// class
Header.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Lanes.Header

See Also

Header Members
MindFusion.Diagramming.Lanes Namespace

Header Members

Represents a header in a lane grid.

The following tables list the members exposed by the Header type.

Public Constructors

	Name	Description
	Header	Initializes a new instance of the **Header** class.

Public Properties

	Name	Description
	height	Gets or sets the height of this header.
	isColumnHeader	Gets a value indicating whether this header is a column header.
	isRowHeader	Gets a value indicating whether this header is a row header.
	resizeType	Gets or sets how a row or a column is resized relatively to its parent.
	rotateTitle	Gets or sets a value indicating whether to rotate the header caption at 90 degrees.
	style	Gets or sets the style of this header.
	subHeaders	Gets a collection of the subheaders of this header.
	title	Gets or sets the header caption.

	width	Gets or sets the width of this header.

Public Methods

	Name	Description
	loadFromXml	Loads the lane header from the specified XML element.
	saveToXml	Serializes the lane header to the specified XML element

See Also

Header Class
MindFusion.Diagramming.Lanes Namespace

Header Constructor

Initializes a new instance of the Header class.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript Copy Code

```
function Header ()
```

See Also

Header Class
MindFusion.Diagramming.Lanes Namespace

Header Methods

Header.loadFromXml Method

Loads the lane header from the specified XML element.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

Header Members
Header Class
MindFusion.Diagramming.Lanes Namespace

Header.saveToXml Method

Serializes the lane header to the specified XML element

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript — Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

Header Members
Header Class
MindFusion.Diagramming.Lanes Namespace

Header Properties

Header.height Property

Gets or sets the height of this header.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript — Copy Code

```
get height() {}
set height(value) {}
```

Property Value

Number

A number specifying the header height.

Remarks

If this is a row header, the property specifies the height of the whole row.

See Also

Header Members
Header Class
MindFusion.Diagramming.Lanes Namespace

Header.isColumnHeader Property

Gets a value indicating whether this header is a column header.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript Copy Code

```
get isColumnHeader() {}
```

Property Value

Boolean

true if the header is a column header, or **false** otherwise.

Remarks

Both **isColumnHeader** and isRowHeader return **false** if the header is not attached to a lane grid.

See Also

Header Members
Header Class
MindFusion.Diagramming.Lanes Namespace

Header.isRowHeader Property

Gets a value indicating whether this header is a row header.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript Copy Code

```
get isRowHeader() {}
```

Property Value

Boolean

true if the header is a row header, or **false** otherwise.

Remarks

Both isColumnHeader and **isRowHeader** return **false** if the header is not attached to a lane grid.

See Also

Header Members
Header Class
MindFusion.Diagramming.Lanes Namespace

Header.resizeType Property

Gets or sets how a row or a column is resized relatively to its parent.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript Copy Code

```
get resizeType() {}
set resizeType(value) {}
```

Property Value

ResizeType

A member of the ResizeType enumeration.

See Also

Header Members
Header Class
MindFusion.Diagramming.Lanes Namespace

Header.rotateTitle Property

Gets or sets a value indicating whether to rotate the header caption at 90 degrees.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript Copy Code

```
get rotateTitle() {}
set rotateTitle(value) {}
```

Property Value

Boolean

true if the title should be rotated, or **false** otherwise.

See Also

Header Members
Header Class
MindFusion.Diagramming.Lanes Namespace

Header.style Property

Gets or sets the style of this header.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript Copy Code

```
get style() {}
set style(value) {}
```

Property Value

Style

A Style object that specifies the appearance of the header.

See Also

Header Members
Header Class
MindFusion.Diagramming.Lanes Namespace

Header.subHeaders Property

Gets a collection of the subheaders of this header.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript Copy Code

```
get subHeaders() {}
```

Property Value

Array

An array containing the subheaders.

Remarks

The returned array should not be modified directly. To add and remove subheaders, use the addColumnHeader, addRowHeader, removeColumnHeader and removeRowHeader methods of the Grid respectively.

See Also

Header Members
Header Class
MindFusion.Diagramming.Lanes Namespace

Header.title Property

Gets or sets the header caption.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript Copy Code

```
get title() {}
set title(value) {}
```

Property Value

String

A string containing the caption text.

See Also

Header Members
Header Class
MindFusion.Diagramming.Lanes Namespace

Header.width Property

Gets or sets the width of this header.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Header.js

Syntax

JavaScript Copy Code

```
get width() {}
set width(value) {}
```

Property Value

Number

A number specifying the header width.

Remarks

If this is a column header, the property specifies the width of the whole column.

See Also

Header Members
Header Class
MindFusion.Diagramming.Lanes Namespace

ResizeType Enumeration

Specifies how to resize child columns and rows in the lanes grid when their parent column or row is resized.

Namespace: MindFusion.Diagramming.Lanes
Assembly: ResizeType.js

Syntax

JavaScript Copy Code

```
// enum
ResizeType = {}
```

Members

	Member name	Description
	Fixed	The header's size (width or height) is fixed and does not change when its parent header is being resized unless all its siblings have fixed size too.
	Proportional	The header resizes in proportion to its parent header.

See Also

MindFusion.Diagramming.Lanes Namespace

Style Class

Encapsulates the appearance properties of an element in a lane grid.

Syntax

JavaScript Copy Code

```
// class
Style.prototype = {}
```

Inheritance Hierarchy

MindFusion.Diagramming.Lanes.Style

See Also

Style Members
MindFusion.Diagramming.Lanes Namespace

Style Members

Encapsulates the appearance properties of an element in a lane grid.

The following tables list the members exposed by the Style type.

Public Constructors

	Name	Description

	Style	Initializes a new instance of the **Style** class.

Public Properties

	Name	Description
	backgroundBrush	Gets or sets the background brush of this style.
	bottomBorderPen	Gets or sets the color used to stoke the bottom border of elements with this style.
	bottomBorderThickness	Gets or sets the width of the bottom border of elements with this style.
	fontName	Gets or sets the name of the font used to draw texts with this style.
	fontSize	Gets or sets the size of the font used to draw texts with this style.
	leftBorderPen	Gets or sets the color used to stoke the left border of elements with this style.
	leftBorderThickness	Gets or sets the width of the left border of elements with this style.
	lineAlignment	Gets or sets a value indicating how to vertically align texts drawn with this style.
	rightBorderPen	Gets or sets the color used to stoke the right border of elements with this style.
	rightBorderThickness	Gets or sets the width of the right border of elements with this style.
	textAlignment	Gets or sets a value indicating how to horizontally align texts drawn with this style.
	textColor	Gets or sets the color used to draw texts with this style.
	topBorderPen	Gets or sets the color used to stoke the top border of elements with this style.
	topBorderThickness	Gets or sets the width of the top border of elements with this style.

Public Methods

	Name	Description
	loadFromXml	Loads the style from the specified XML element
	saveToXml	Serializes the lane style to the specified XML element

See Also

Style Class
MindFusion.Diagramming.Lanes Namespace

Style Constructor

Initializes a new instance of the Style class.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
function Style ()
```

See Also

Style Methods

Style.loadFromXml Method

Loads the style from the specified XML element

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
function loadFromXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.saveToXml Method

Serializes the lane style to the specified XML element

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
function saveToXml (xmlElement, context)
```

Parameters

xmlElement
Element. An XML DOM element that will contain the item's data.

context
XmlPersistContext. An object providing contextual information about the serialization process and some helper serialization methods.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style Properties

Style.backgroundBrush Property

Gets or sets the background brush of this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get backgroundBrush() {}
set backgroundBrush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.bottomBorderPen Property

Gets or sets the color used to stoke the bottom border of elements with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get bottomBorderPen() {}
set bottomBorderPen(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.bottomBorderThickness Property

Gets or sets the width of the bottom border of elements with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get bottomBorderThickness() {}
set bottomBorderThickness(value) {}
```

Property Value

Number

A number specifying the border width.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.fontName Property

Gets or sets the name of the font used to draw texts with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get fontName() {}
set fontName(value) {}
```

Property Value

String

A string specifying font name.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.fontSize Property

Gets or sets the size of the font used to draw texts with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get fontSize() {}
set fontSize(value) {}
```

Property Value

Number

A number specifying font size in current measure unit.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.leftBorderPen Property

Gets or sets the color used to stoke the left border of elements with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get leftBorderPen() {}
set leftBorderPen(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.leftBorderThickness Property

Gets or sets the width of the left border of elements with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get leftBorderThickness() {}
set leftBorderThickness(value) {}
```

Property Value

Number

A number specifying the border width.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.lineAlignment Property

Gets or sets a value indicating how to vertically align texts drawn with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get lineAlignment() {}
set lineAlignment(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

Style.rightBorderPen Property

Gets or sets the color used to stoke the right border of elements with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get rightBorderPen() {}
set rightBorderPen(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.rightBorderThickness Property

Gets or sets the width of the right border of elements with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get rightBorderThickness() {}
set rightBorderThickness(value) {}
```

Property Value

Number

A number specifying the border width.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.textAlignment Property

Gets or sets a value indicating how to horizontally align texts drawn with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get textAlignment() {}
set textAlignment(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.textColor Property

Gets or sets the color used to draw texts with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get textColor() {}
set textColor(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.topBorderPen Property

Gets or sets the color used to stoke the top border of elements with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript Copy Code

```
get topBorderPen() {}
set topBorderPen(value) {}
```

Property Value

String

A string specifying the color name or hexadecimal value.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

Style.topBorderThickness Property

Gets or sets the width of the top border of elements with this style.

Namespace: MindFusion.Diagramming.Lanes
Assembly: Style.js

Syntax

JavaScript — Copy Code

```
get topBorderThickness() {}
set topBorderThickness(value) {}
```

Property Value

Number

A number specifying the border width.

See Also

Style Members
Style Class
MindFusion.Diagramming.Lanes Namespace

MindFusion.Drawing Namespace

Alignment Enumeration

Specifies the alignment of a graphics element relative to its layout rectangle.

Namespace: MindFusion.Drawing
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
Alignment = {}
```

Members

	Member name	Description
	Near	The element is drawn at the near edge of the layout rectangle (left or top).
	Center	The element is drawn at the center of the layout rectangle.
	Far	The element is drawn at the far edge of the layout rectangle (right or bottom).

See Also

MindFusion.Drawing Namespace

Border3D Class

Represents a 3D border.

Syntax

JavaScript — Copy Code

```
// class
Border3D.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.Border3D

See Also

Border3D Members
MindFusion.Drawing Namespace

Border3D Members

Represents a 3D border.

The following tables list the members exposed by the Border3D type.

Public Constructors

	Name	Description
	Border3D	Initializes a new instance of the Border3D class.

See Also

Border3D Class
MindFusion.Drawing Namespace

Border3D Constructor

Initializes a new instance of the Border3D class.

Namespace: MindFusion.Drawing
Assembly: Border3D.js

Syntax

JavaScript Copy Code

```
function Border3D (rect)
```

Parameters

rect
Rect. A Rect instance containing the border coordinates.

See Also

Border3D Class
MindFusion.Drawing Namespace

Component Class

Represents components in CompositeNode visual tree.

Syntax

JavaScript Copy Code

```
// class
Component.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.ComponentBase
MindFusion.Drawing.Component
MindFusion.Diagramming.Button
MindFusion.Drawing.Container

See Also

Component Members
MindFusion.Drawing Namespace

Component Members

Represents components in CompositeNode visual tree.

The following tables list the members exposed by the Component type.

Public Constructors

	Name	Description
	Component	Initializes a new instance of the **Component** class.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

See Also

Component Class
MindFusion.Drawing Namespace

Component Constructor

Initializes a new instance of the Component class.

Namespace: MindFusion.Drawing
Assembly: Component.js

Syntax

JavaScript Copy Code

```
function Component ()
```

See Also

Component Class
MindFusion.Drawing Namespace

ComponentBase Class

A base class for components in CompositeNode visual tree.

Syntax

JavaScript — Copy Code

```
// class
ComponentBase.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.ComponentBase
MindFusion.Drawing.Component
MindFusion.Drawing.Ellipse
MindFusion.Drawing.Image
MindFusion.Drawing.Path
MindFusion.Drawing.Rect
MindFusion.Drawing.Text
MindFusion.Drawing.Video

See Also

ComponentBase Members
MindFusion.Drawing Namespace

ComponentBase Members

A base class for components in CompositeNode visual tree.

The following tables list the members exposed by the ComponentBase type.

Public Constructors

	Name	Description
	ComponentBase	Initializes a new instance of the **ComponentBase** class.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in.
	gridRow	Gets or sets the index of the grid row this component is contained in.
	horizontalAlignment	Gets or sets the component's horizontal alignment.
	verticalAlignment	Gets or sets the component's vertical alignment.
	visibility	Gets or sets the visibility of this component.

See Also

ComponentBase Class
MindFusion.Drawing Namespace

ComponentBase Constructor

Initializes a new instance of the ComponentBase class.

Namespace: MindFusion.Drawing
Assembly: ComponentBase.js

Syntax

JavaScript Copy Code

```
function ComponentBase ()
```

See Also

ComponentBase Class
MindFusion.Drawing Namespace

ComponentBase Methods

ComponentBase Properties

ComponentBase.gridColumn Property

Gets or sets the index of the grid column this component is contained in.

Namespace: MindFusion.Drawing
Assembly: ComponentBase.js

Syntax

JavaScript Copy Code

```
get gridColumn() {}
set gridColumn(value) {}
```

Property Value

Number

The grid column index.

Remarks

This value is used when the component is placed inside a GridPanel.

See Also

ComponentBase Members
ComponentBase Class
MindFusion.Drawing Namespace

ComponentBase.gridRow Property

Gets or sets the index of the grid row this component is contained in.

Namespace: MindFusion.Drawing
Assembly: ComponentBase.js

Syntax

JavaScript Copy Code

```
get gridRow() {}
set gridRow(value) {}
```

Property Value

Number

The grid row index.

Remarks

This value is used when the component is placed inside a GridPanel.

See Also

ComponentBase Members
ComponentBase Class
MindFusion.Drawing Namespace

ComponentBase.horizontalAlignment Property

Gets or sets the component's horizontal alignment.

Namespace: MindFusion.Drawing
Assembly: ComponentBase.js

Syntax

JavaScript Copy Code

```
get horizontalAlignment() {}
set horizontalAlignment(value) {}
```

Property Value

LayoutAlignment

A member of the LayoutAlignment enumeration.

See Also

ComponentBase Members
ComponentBase Class
MindFusion.Drawing Namespace

ComponentBase.verticalAlignment Property

Gets or sets the component's vertical alignment.

Namespace: MindFusion.Drawing
Assembly: ComponentBase.js

Syntax

JavaScript Copy Code

```
get verticalAlignment() {}
set verticalAlignment(value) {}
```

Property Value

LayoutAlignment

A member of the LayoutAlignment enumeration.

See Also

ComponentBase Members
ComponentBase Class
MindFusion.Drawing Namespace

ComponentBase.visibility Property

Gets or sets the visibility of this component.

Namespace: MindFusion.Drawing
Assembly: ComponentBase.js

Syntax

JavaScript — Copy Code

```
get visibility() {}
set visibility(value) {}
```

Property Value

Visibility

A member of the Visibility enumeration.

See Also

ComponentBase Members
ComponentBase Class
MindFusion.Drawing Namespace

Container Class

Represents container components in drawing tree.

Syntax

JavaScript — Copy Code

```
// class
Container.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.ComponentBase
  MindFusion.Drawing.Component
    MindFusion.Drawing.Container
      MindFusion.Diagramming.Manipulator
      MindFusion.Diagramming.Panel

See Also

Container Members
MindFusion.Drawing Namespace

Container Members

Represents container components in drawing tree.

The following tables list the members exposed by the Container type.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

See Also

Container Class
MindFusion.Drawing Namespace

DashStyle Enumeration

Specifies the dash pattern of lines.

Namespace: MindFusion.Drawing
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
DashStyle = {}
```

Members

	Member name	Description
	Solid	Draw solid line.
	Dash	Draw dashed line.
	Dot	Draw dotted line.
	DashDot	Draw dash-dot pattern.
	DashDotDot	Draw dash-dot-dot pattern.
	Custom	Custom pattern.

See Also

MindFusion.Drawing Namespace

DrawingUtils Class

Contains drawing-related helper functions.

Syntax

JavaScript — Copy Code

```
// class
DrawingUtils.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.DrawingUtils

See Also

DrawingUtils Members
MindFusion.Drawing Namespace

DrawingUtils Members

Contains drawing-related helper functions.

The following tables list the members exposed by the DrawingUtils type.

Public Methods

	Name	Description
	approximateBezier	Approximates a Bezier curve with the given quality and by a series of line segments.
	checkIntersect	Checks if the specified rectangle and circle intersect.
	degrees	Converts the specified value from radians to degrees.
	distToPolyline	Calculates the shortest distance from the specified point to the specified polyline, also returning the index of the segment the point is closest to.
	distToRectPoint	Returns the point that lies on the outline of a rectangle and is nearest to the given point.
	getCenter	Returns the center of the specified rectangle.
	getPointsBounds	Calculates the bounding rect of the specified polyline.
	getSegmentIntersection	Checks whether the segments defined by the specified point pairs intersect and returns the intersection point.
	inflate	Creates and returns an enlarged copy of a Rect instance.
	intersect	Determines whether the specified line segments intersect.
	minDistToRect	Calculates the minimum distance between a given point and a given rectangle.
	pointInCircle	Checks whether the specified circle contains the specified point.
	pointInEllipse	Checks whether the specified ellipse contains the specified point.
	pointInPolygon	Checks whether the specified rectangle contains the specified point.
	radians	Converts the specified value from degrees to radians.
	unionRects	Returns the smallest possible rectangle containing both of the specified rectangles.

See Also

DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils Methods

DrawingUtils.approximateBezier Method

Approximates a Bezier curve with the given quality and by a series of line segments.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function approximateBezier (points, quality, start)
```

Parameters

points
Array. A list of Bezier control points.

quality
Number. Specifies the approximation quality.

start
Number. Specifies the start point index.

Return Value
Array. A list of points that approximate the curve as a series of straight line segments.

See Also
DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.checkIntersect Method

Checks if the specified rectangle and circle intersect.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function checkIntersect (point, rect, rad)
```

Parameters
point
Point. The center of the circle.

rect
Rect. The rectangle.

rad
Number. The radius of the circle.

Return Value
Boolean. true if the rectangle and the circle intersect; otherwise, false.

See Also
DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.degrees Method

Converts the specified value from radians to degrees.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function degrees (radians)
```

Parameters
radians
Number. The radians to convert.

Return Value
Number. The converted angle.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.distToPolyline Method

Calculates the shortest distance from the specified point to the specified polyline, also returning the index of the segment the point is closest to.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function distToPolyline ()
```

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.distToRectPoint Method

Returns the point that lies on the outline of a rectangle and is nearest to the given point.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function distToRectPoint (point, rect)
```

Parameters

point
Point. The point to consider.

rect
Rect. The rectangle to consider.

Return Value

Number. The nearest point.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.getCenter Method

Returns the center of the specified rectangle.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function getCenter (rect)
```

Parameters

rect
Rect. The rectangle.

Return Value

Point. The center.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.getPointsBounds Method

Calculates the bounding rect of the specified polyline.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function getPointsBounds (points)
```

Parameters

points
Array. An array of points.

Return Value

Rect. The bounding rect.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.getSegmentIntersection Method

Checks whether the segments defined by the specified point pairs intersect and returns the intersection point.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function getSegmentIntersection (p1, p2, p3, p4)
```

Parameters

p1
Point. The start of the first segment.

p2
Point. The end of the first segment.

p3
Point. The start of the second segment.

p4

Point. The end of the second segment.

Return Value

Boolean. true if the segments intersect, false if not.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.inflate Method

Creates and returns an enlarged copy of a Rect instance.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function inflate (rect, x, y)
```

Parameters

rect
Rect. The Rect instance to inflate.

x
Number. The amount to inflate horizontally.

y
Number. The amount to inflate vertically.

Return Value

Rect. The inflated Rect.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.intersect Method

Determines whether the specified line segments intersect.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function intersect (p1, p2, p3, p4)
```

Parameters

p1
Point. The start of the first line segment.

p2
Point. The end of the first line segment.

p3
Point. The start of the second line segment.

p4
Point. The end of the second line segment.

Return Value

Boolean. true if the segments intersect, false if not.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.minDistToRect Method

Calculates the minimum distance between a given point and a given rectangle.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function minDistToRect (point, rect)
```

Parameters

point
Point. The point.

rect
Rectabgle. The rectangle.

Return Value

Number. The minimum distance.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.pointInCircle Method

Checks whether the specified circle contains the specified point.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function pointInCircle (point, center, radius)
```

Parameters

point
Point. The point to check.

center
Point. The center of the circle.

radius
Number. The radius of the circle.

Return Value

Boolean. true if the circle contains the specified point, false if not.

See Also

DrawingUtils.pointInEllipse Method

Checks whether the specified ellipse contains the specified point.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function pointInEllipse (point, rect)
```

Parameters

point
Point. The point to check.

rect
Rect. The bounds of the ellipse.

Return Value

Boolean. true if the ellipse contains the specified point, false if not.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.pointInPolygon Method

Checks whether the specified rectangle contains the specified point.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript Copy Code

```
function pointInPolygon (point, polygon)
```

Parameters

point
Point. The point to check.

polygon
Array. An array with the points that define the polygon.

Return Value

Boolean. true if the polygon contains the specified point, false if not.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.radians Method

Converts the specified value from degrees to radians.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript — Copy Code

```
function radians (degrees)
```

Parameters

degrees
Number. The degrees to convert.

Return Value

Number. The converted angle.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

DrawingUtils.unionRects Method

Returns the smallest possible rectangle containing both of the specified rectangles.

Namespace: MindFusion.Drawing
Assembly: DrawingUtils.js

Syntax

JavaScript — Copy Code

```
function unionRects (rect1, rect2)
```

Parameters

rect1
Rect. The first rectangle.

rect2
Rect. The second rectangle.

Return Value

Rect. A Rect instance that represents the union of the specified arguments.

See Also

DrawingUtils Members
DrawingUtils Class
MindFusion.Drawing Namespace

Ellipse Class

Represents an Ellipse.

Syntax

JavaScript — Copy Code

```
// class
Ellipse.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.ComponentBase
MindFusion.Drawing.Ellipse

See Also

Ellipse Members
MindFusion.Drawing Namespace

Ellipse Members

Represents an Ellipse.

The following tables list the members exposed by the Ellipse type.

Public Constructors

	Name	Description
	Ellipse	Initializes a new instance of the Ellipse class.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	height	Gets or sets the height of the ellipse.
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)
	width	Gets or sets the width of the ellipse.
	x	Gets or sets the x-coordinate of the upper-left corner of the ellipse.
	y	Gets or sets the y-coordinate of the upper-left corner of the ellipse.

Public Methods

	Name	Description
	clone	Creates a **Ellipse** object identical to the current object.

See Also

Ellipse Class
MindFusion.Drawing Namespace

Ellipse Constructor

Initializes a new instance of the Ellipse class.

Namespace: MindFusion.Drawing
Assembly: Ellipse.js

Syntax

JavaScript Copy Code

```
function Ellipse (x, y, width, height)
```

Parameters

x
Number. The X-coordinate of the top left corner of the Ellipse.

y
Number. The Y-coordinate of the top left corner of the Ellipse.

width
Number. The width of the Ellipse.

height
Number. The height of the Ellipse.

See Also

Ellipse Class
MindFusion.Drawing Namespace

Ellipse Methods

Ellipse.clone Method

Creates a Ellipse object identical to the current object.

Namespace: MindFusion.Drawing
Assembly: Ellipse.js

Syntax

JavaScript Copy Code

```
function clone ()
```

Return Value

The newly created Ellipse object.

See Also

Ellipse Members
Ellipse Class
MindFusion.Drawing Namespace

Ellipse Properties

Ellipse.height Property

Gets or sets the height of the ellipse.

Namespace: MindFusion.Drawing
Assembly: Ellipse.js

Syntax

JavaScript Copy Code

```
get height() {}
set height(value) {}
```

Property Value

Number

The height of the ellipse.

See Also

Ellipse Members
Ellipse Class
MindFusion.Drawing Namespace

Ellipse.width Property

Gets or sets the width of the ellipse.

Namespace: MindFusion.Drawing
Assembly: Ellipse.js

Syntax

JavaScript Copy Code

```
get width() {}
set width(value) {}
```

Property Value

Number

The width of the ellipse.

See Also

Ellipse Members
Ellipse Class
MindFusion.Drawing Namespace

Ellipse.x Property

Gets or sets the x-coordinate of the upper-left corner of the ellipse.

Namespace: MindFusion.Drawing
Assembly: Ellipse.js

Syntax

JavaScript Copy Code

```
get x() {}
set x(value) {}
```

Property Value

Number

The x-coordinate of the upper-left corner of the ellipse.

See Also

Ellipse Members
Ellipse Class
MindFusion.Drawing Namespace

Ellipse.y Property

Gets or sets the y-coordinate of the upper-left corner of the ellipse.

Namespace: MindFusion.Drawing
Assembly: Ellipse.js

Syntax

JavaScript — Copy Code

```
get y() {}
set y(value) {}
```

Property Value

Number

The x-coordinate of the upper-left corner of the ellipse.

See Also

Ellipse Members
Ellipse Class
MindFusion.Drawing Namespace

Font Class

Represents a font.

Syntax

JavaScript — Copy Code

```
// class
Font.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.Font

See Also

Font Members
MindFusion.Drawing Namespace

Font Members

Represents a font.

The following tables list the members exposed by the Font type.

Public Constructors

	Name	Description
	Font	Initializes a new instance of the Font class with the specified attributes.

Public Properties

	Name	Description
	bold	Gets whether the font is bold.
	fontStyle	Gets or sets the style of this font.
	italic	Gets whether the font is italic.
	name	Gets or sets the name of this font.
	size	Gets or sets the size of this font.
	underline	Gets whether the font is underlined.

Public Methods

	Name	Description
	apply	Applies the font properties to a CanvasRenderingContext2D drawing object.
S	fromObject	Returns a Font from a JSON object or string describing this font.
	toObject	Returns a JSON object describing this font.

See Also

Font Class
MindFusion.Drawing Namespace

Font Constructor

Initializes a new instance of the Font class with the specified attributes.

Namespace: MindFusion.Drawing
Assembly: Font.js

Syntax

JavaScript — Copy Code

```
function Font (name, [size, [bold, [italic, [underline]]]])
```

Parameters

name
String. The font name.

size
Optional.
Number. The font size.

bold
Optional.
Boolean | FontStyle. true if this font is bold, or false otherwise.

italic
Optional.
Boolean. true if this font is italic, or false otherwise.

underline
Optional.
Boolean. true if this font is underlined, or false otherwise.

See Also

Font Class
MindFusion.Drawing Namespace

Font Methods

Font.apply Method

Applies the font properties to a CanvasRenderingContext2D drawing object.

Namespace: MindFusion.Drawing
Assembly: Font.js

Syntax

JavaScript Copy Code

```
function apply (context)
```

Parameters

context
CanvasRenderingContext2D. The CanvasRenderingContext2D drawing object.

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.fromObject Method

Returns a Font from a JSON object or string describing this font.

Namespace: MindFusion.Drawing
Assembly: Font.js

Syntax

JavaScript Copy Code

```
function fromObject ()
```

Return Value

Font. The font object.

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.toObject Method

Returns a JSON object describing this font.

Namespace: MindFusion.Drawing
Assembly: Font.js

Syntax

JavaScript Copy Code

```
function toObject ()
```

Return Value

Object. The object describing this font.

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font Properties

Font.bold Property

Gets whether the font is bold.

Namespace: MindFusion.Drawing
Assembly: Font.js

Syntax

JavaScript — Copy Code

```
get bold() {}
```

Property Value

Boolean

true if Bold flag is set in fontStyle, or **false** otherwise.

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.fontStyle Property

Gets or sets the style of this font.

Namespace: MindFusion.Drawing
Assembly: Font.js

Syntax

JavaScript — Copy Code

```
get fontStyle() {}
set fontStyle(value) {}
```

Property Value

FontStyle

A combination of FontStyle enumeration flags.

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.italic Property

Gets whether the font is italic.

Namespace: MindFusion.Drawing
Assembly: Font.js

Syntax

JavaScript — Copy Code

```
get italic() {}
```

Property Value

Boolean

true if Italic flag is set in fontStyle, or **false** otherwise.

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.name Property

Gets or sets the name of this font.

Namespace: MindFusion.Drawing
Assembly: Font.js

Syntax

JavaScript Copy Code

```
get name() {}
set name(value) {}
```

Property Value

String

A string containing font name.

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.size Property

Gets or sets the size of this font.

Namespace: MindFusion.Drawing
Assembly: Font.js

Syntax

JavaScript Copy Code

```
get size() {}
set size(value) {}
```

Property Value

Number

The font size.

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.underline Property

Gets whether the font is underlined.

Namespace: MindFusion.Drawing
Assembly: Font.js

Syntax

JavaScript Copy Code

```
get underline() {}
```

Property Value

Boolean

true if Underline flag is set in fontStyle, or **false** otherwise.

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

FontStyle Enumeration

Specifies font style attributes.

Namespace: MindFusion.Drawing
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
FontStyle = {}
```

Members

	Member name	Description
	Regular	Normal text.
	Bold	Bold text.
	Italic	Italic text.
	Underline	Underlined text.
	Strikeout	Striked text.

See Also

MindFusion.Drawing Namespace

GraphicsUnit Class

Specifies the unit of measure for length and size properties.

Syntax

JavaScript Copy Code

```
// class
GraphicsUnit.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.GraphicsUnit

See Also

GraphicsUnit Members

Specifies the unit of measure for length and size properties.

The following tables list the members exposed by the GraphicsUnit type.

Public Properties

	Name	Description
	Centimeter	Specifies the centimeter as the unit of measure.
	Display	Specifies the display unit as the unit of measure.
	Document	Specifies the document unit as the unit of measure.
	Inch	Specifies the inch as the unit of measure.
	Millimeter	Specifies the millimeter as the unit of measure.
	Percent	Specifies the world coordinate system unit as the unit of measure.
	Pixel	Specifies the pixel unit as the unit of measure.
	Point	Specifies a printer's point as the unit of measure.
	World	Specifies the world coordinate system unit as the unit of measure.
	WpfPoint	Specifies the WpfPoint unit as the unit of measure.

See Also

GraphicsUnit Class
MindFusion.Drawing Namespace

GraphicsUnit Properties

GraphicsUnit.Centimeter Property

Specifies the centimeter as the unit of measure.

Namespace: MindFusion.Drawing
Assembly: GraphicsUnit.js

Syntax

JavaScript Copy Code

```
get Centimeter() {}
set Centimeter(value) {}
```

Property Value

Number

Defines 2.54 DPI resolution.

See Also

GraphicsUnit Members
GraphicsUnit Class
MindFusion.Drawing Namespace

GraphicsUnit.Display Property

Specifies the display unit as the unit of measure.

Namespace: MindFusion.Drawing
Assembly: GraphicsUnit.js

Syntax

JavaScript Copy Code

```
get Display() {}
set Display(value) {}
```

Property Value

Number

Defines 100 DPI resolution.

See Also

GraphicsUnit Members
GraphicsUnit Class
MindFusion.Drawing Namespace

GraphicsUnit.Document Property

Specifies the document unit as the unit of measure.

Namespace: MindFusion.Drawing
Assembly: GraphicsUnit.js

Syntax

JavaScript Copy Code

```
get Document() {}
set Document(value) {}
```

Property Value

Number

Defines 300 DPI resolution.

See Also

GraphicsUnit Members
GraphicsUnit Class
MindFusion.Drawing Namespace

GraphicsUnit.Inch Property

Specifies the inch as the unit of measure.

Namespace: MindFusion.Drawing
Assembly: GraphicsUnit.js

Syntax

JavaScript Copy Code

```
get Inch() {}
set Inch(value) {}
```

Property Value

Number

Defines 1 DPI resolution.

See Also

GraphicsUnit Members
GraphicsUnit Class
MindFusion.Drawing Namespace

GraphicsUnit.Millimeter Property

Specifies the millimeter as the unit of measure.

Namespace: MindFusion.Drawing
Assembly: GraphicsUnit.js

Syntax

JavaScript Copy Code

```
get Millimeter() {}
set Millimeter(value) {}
```

Property Value

Number

Defines 25.4 DPI resolution.

See Also

GraphicsUnit Members
GraphicsUnit Class
MindFusion.Drawing Namespace

GraphicsUnit.Percent Property

Specifies the world coordinate system unit as the unit of measure.

Namespace: MindFusion.Drawing
Assembly: GraphicsUnit.js

Syntax

JavaScript Copy Code

```
get Percent() {}
set Percent(value) {}
```

Property Value

Number

For internal use.

See Also

GraphicsUnit Members
GraphicsUnit Class
MindFusion.Drawing Namespace

GraphicsUnit.Pixel Property

Specifies the pixel unit as the unit of measure.

Namespace: MindFusion.Drawing
Assembly: GraphicsUnit.js

Syntax

JavaScript Copy Code

```
get Pixel() {}
set Pixel(value) {}
```

Property Value

Number

Defines 96 DPI resolution.

See Also

GraphicsUnit Members
GraphicsUnit Class
MindFusion.Drawing Namespace

GraphicsUnit.Point Property

Specifies a printer's point as the unit of measure.

Namespace: MindFusion.Drawing
Assembly: GraphicsUnit.js

Syntax

JavaScript Copy Code

```
get Point() {}
set Point(value) {}
```

Property Value

Number

Defines 72 DPI resolution.

See Also

GraphicsUnit Members
GraphicsUnit Class
MindFusion.Drawing Namespace

GraphicsUnit.World Property

Specifies the world coordinate system unit as the unit of measure.

Namespace: MindFusion.Drawing
Assembly: GraphicsUnit.js

Syntax

JavaScript Copy Code

```
get World() {}
set World(value) {}
```

Property Value

Number

For internal use.

See Also

GraphicsUnit Members
GraphicsUnit Class
MindFusion.Drawing Namespace

GraphicsUnit.WpfPoint Property

Specifies the WpfPoint unit as the unit of measure.

Namespace: MindFusion.Drawing
Assembly: GraphicsUnit.js

Syntax

JavaScript Copy Code

```
get WpfPoint() {}
set WpfPoint(value) {}
```

Property Value

Number

Defines 96 DPI resolution.

See Also

GraphicsUnit Members
GraphicsUnit Class
MindFusion.Drawing Namespace

Image Class

Represents an image.

Syntax

JavaScript Copy Code

```
// class
Image.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.ComponentBase
MindFusion.Drawing.Image

See Also

Image Members
MindFusion.Drawing Namespace

Image Members

Represents an image.

The following tables list the members exposed by the Image type.

Public Constructors

	Name	Description
	Image	Initializes a new instance of the **Image** class.

Public Fields

	Name	Description
	bounds	The object's bounds.
	image	The image to be displayed.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

See Also

Image Class
MindFusion.Drawing Namespace

Image Fields

Image.bounds Field

The object's bounds.

Namespace: MindFusion.Drawing
Assembly: Image.js

Syntax

JavaScript Copy Code

```
var bounds
```

See Also

Image Members
Image Class
MindFusion.Drawing Namespace

Image.image Field

The image to be displayed.

Namespace: MindFusion.Drawing
Assembly: Image.js

Syntax

JavaScript Copy Code

```
var image
```

See Also

Image Members
Image Class
MindFusion.Drawing Namespace

Image Constructor

Initializes a new instance of the Image class.

Namespace: MindFusion.Drawing
Assembly: Image.js

Syntax

JavaScript — Copy Code

```
function Image (bounds)
```

Parameters

bounds
Rect. The bounds of the image.

See Also

Image Class
MindFusion.Drawing Namespace

ImageAlign Enumeration

Specifies the position and alignment of a picture in a node, or that of the background image.

Namespace: MindFusion.Drawing
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
ImageAlign = {}
```

Members

	Member name	Description
	Center	The image is centered in the node or diagram.
	Fit	The image is resized to fit the size of the object or the component's client area.
	Stretch	The image is stretched to fill the object or the component's client area.
	Tile	The image is tiled to cover the node or the diagram.
	TopLeft	The image is aligned to the top left corner of the node or the diagram.
	BottomLeft	The image is aligned to the bottom left corner of the node or the diagram.
	TopRight	The image is aligned to the top right corner of the node or the diagram.
	BottomRight	The image is aligned to the bottom right corner of the node or the diagram.
	TopCenter	The image is centered horizontally and aligned to the top side.

	BottomCenter	The image is centered horizontally and aligned to the bottom side.
	MiddleLeft	The image is centered vertically and aligned to the left-hand side.
	MiddleRight	The image is centered vertically and aligned to the right-hand side.
	FitLeft	The image is resized to fit the size of the object and aligned to left side.
	FitTop	The image is resized to fit the size of the object and aligned to top side.
	FitRight	The image is resized to fit the size of the object and aligned to right side.
	FitBottom	The image is resized to fit the size of the object and aligned to bottom side.

See Also

MindFusion.Drawing Namespace

LayoutAlignment Enumeration

Specifies alignment of components.

Namespace: MindFusion.Drawing
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
LayoutAlignment = {}
```

Members

	Member name	Description
	Near	Left or top alignment.
	Center	Center alignment.
	Far	Right ot bottom alignment
	Stretch	Stretch the component.

See Also

MindFusion.Drawing Namespace

LineJoin Enumeration

Specifies how to join consecutive line or curve segments in a figure.

Namespace: MindFusion.Drawing
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
LineJoin = {}
```

Members

	Member name	Description
	Miter	Produces a sharp corner or a clipped corner, depending on whether the length of the miter exceeds the miter limit.
	Bevel	Produces a diagonal corner.
	Round	Produces a smooth, circular arc between the lines.

See Also

MindFusion.Drawing Namespace

Path Class

Represents a path.

Syntax

JavaScript Copy Code

```
// class
Path.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.ComponentBase
  MindFusion.Drawing.Path

See Also

Path Members
MindFusion.Drawing Namespace

Path Members

Represents a path.

The following tables list the members exposed by the Path type.

Public Constructors

	Name	Description
	Path	Initializes a new instance of the **Path** class.

Public Properties

	Name	Description
	brush	Gets or sets the brush used to fill a closed path.
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	pen	Gets or sets the pen used to draw the path.
	text	Gets or sets the text displayed inside this path.

	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

Public Methods

	Name	Description
	addEllipse	Adds an ellipse figure to the path.
	addRect	Adds a rectangle figure to the path.
	arcTo	Draws an arc.
	bezierTo	Draws a cubic bezier curve.
	clone	Creates a **Path** object identical to the current object.
	close	Closes the path.
	done	Ends the path definition.
	empty	Gets a value indicating whether this **Path** is empty.
	getBounds	Gets the bounding rect of the path.
	init	Begins a path or resets the current path.
	lineTo	Draws a line from the current point to the specified location.
	moveTo	Moves the path to the specified location.
	quadraticCurveTo	Draws a quadratic bezier curve.

See Also

Path Class
MindFusion.Drawing Namespace

Path Constructor

Initializes a new instance of the Path class.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript Copy Code

```
function Path ([pathString])
```

Parameters

pathString
Optional.
String. A string representing the path figures.

See Also

Path Class
MindFusion.Drawing Namespace

Path Methods

Path.addEllipse Method

Adds an ellipse figure to the path.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript — Copy Code

```
function addEllipse (x, y, width, height)
```

Parameters

x
Number. The x-coordinate of the center of the ellipse.

y
Number. The y-coordinate of the center of the ellipse.

width
Number. The width of the ellipse.

height
Number. The height of the ellipse.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.addRect Method

Adds a rectangle figure to the path.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript — Copy Code

```
function addRect (x, y, width, height)
```

Parameters

x
Number. The x-coordinate of the rectangle.

y
Number. The y-coordinate of the rectangle.

width
Number. The width of the rectangle.

height
Number. The height of the rectangle.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.arcTo Method

Draws an arc.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript — Copy Code

```
function arcTo (x, y, radius, startAngle, endAngle, anticlockwise)
```

Parameters

x
Number. The x-coordinate of the center of the circle.

y
Number. The y-coordinate of the center of the circle.

radius
Number. The radius of the circle

startAngle
Number. The starting angle in radians.

endAngle
Number. The ending angle in radians.

anticlockwise
Boolean. Specifies whether the drawing should be counterclockwise or clockwise.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.bezierTo Method

Draws a cubic bezier curve.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript — Copy Code

```
function bezierTo (x1, y1, x2, y2, x3, y3)
```

Parameters

x1
Number. The x-coordinate of the first bezier control point.

y1
Number. The y-coordinate of the first bezier control point.

x2
Number. The x-coordinate of the second bezier control point.

y2
Number. The y-coordinate of the second bezier control point.

x3
Number. The x-coordinate of the ending point.

y3
Number. The y-coordinate of the ending point.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.clone Method

Creates a Path object identical to the current object.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript Copy Code

```
function clone ()
```

Return Value

The newly created Path object.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.close Method

Closes the path.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript Copy Code

```
function close ()
```

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.done Method

Ends the path definition.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript Copy Code

```
function done ()
```

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.empty Method

Gets a value indicating whether this Path is empty.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript Copy Code

```
function empty ()
```

Return Value

true if the path does not contain any figures; otherwise, **false**.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.getBounds Method

Gets the bounding rect of the path.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript Copy Code

```
function getBounds ()
```

Return Value

Rect. The bounding rect.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.init Method

Begins a path or resets the current path.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript Copy Code

```
function init ()
```

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.lineTo Method

Draws a line from the current point to the specified location.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript — Copy Code

```
function lineTo (x, y)
```

Parameters

x
Number. The x-coordinate of the point.

y
Number. The y-coordinate of the point.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.moveTo Method

Moves the path to the specified location.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript — Copy Code

```
function moveTo (x, y)
```

Parameters

x
Number. The x-coordinate of the point.

y
Number. The y-coordinate of the point.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.quadraticCurveTo Method

Draws a quadratic bezier curve.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript — Copy Code

```
function quadraticCurveTo (x1, y1, x, y)
```

Parameters

x1
Number. The x-coordinate of the bezier control point.

y1
Number. The y-coordinate of the bezier control point.

x
Number. The x-coordinate of the ending point.

y
Number. The y-coordinate of the ending point.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path Properties

Path.brush Property

Gets or sets the brush used to fill a closed path.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript Copy Code

```
get brush() {}
set brush(value) {}
```

Property Value

Object

A color string or an object literal specifying gradient.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.pen Property

Gets or sets the pen used to draw the path.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript Copy Code

```
get pen() {}
set pen(value) {}
```

Property Value

Object

A color string.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Path.text Property

Gets or sets the text displayed inside this path.

Namespace: MindFusion.Drawing
Assembly: Path.js

Syntax

JavaScript Copy Code

```
get text() {}
set text(value) {}
```

Property Value

String

A string containing path's text.

See Also

Path Members
Path Class
MindFusion.Drawing Namespace

Point Class

Represents a point.

Syntax

JavaScript Copy Code

```
// class
Point.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.Point

See Also

Point Members
MindFusion.Drawing Namespace

Point Members

Represents a point.

The following tables list the members exposed by the Point type.

Public Constructors

	Name	Description
	Point	Initializes a new instance of the **Point** class with the specified X and Y coordinates.

Public Properties

	Name	Description
	x	Gets or sets the x-coordinate of the point.
	y	Gets or sets the y-coordinate of the point.

Public Methods

	Name	Description
	empty	Gets a value indicating whether this **Point** is empty.
	equals	Specifies whether this **Point** contains the same coordinates as the specified point.

See Also

Point Class
MindFusion.Drawing Namespace

Point Constructor

Initializes a new instance of the Point class with the specified X and Y coordinates.

Namespace: MindFusion.Drawing
Assembly: Point.js

Syntax

JavaScript Copy Code

```
function Point (x, y)
```

Parameters

x
The X-coordinate of the Point.

y
The Y-coordinate of the **Point**.

See Also

Point Class
MindFusion.Drawing Namespace

Point Methods

Point.empty Method

Gets a value indicating whether this Point is empty.

Namespace: MindFusion.Drawing
Assembly: Point.js

Syntax

JavaScript Copy Code

```
function empty ()
```

Return Value

true if both x and y are 0; otherwise, **false**.

See Also

Point Members
Point Class
MindFusion.Drawing Namespace

Point.equals Method

Specifies whether this Point contains the same coordinates as the specified point.

Namespace: MindFusion.Drawing
Assembly: Point.js

Syntax

JavaScript Copy Code

```
function equals ([point])
```

Parameters

point
Optional. The Point to test.

Return Value

true if *point* is a Point and has the same coordinates as this **Point**.

See Also

Point Members
Point Class
MindFusion.Drawing Namespace

Point Properties

Point.x Property

Gets or sets the x-coordinate of the point.

Namespace: MindFusion.Drawing
Assembly: Point.js

Syntax

JavaScript Copy Code

```
get x() {}
set x(value) {}
```

Property Value

Number

The x-coordinate of the point.

See Also

Point Members
Point Class
MindFusion.Drawing Namespace

Point.y Property

Gets or sets the y-coordinate of the point.

Namespace: MindFusion.Drawing
Assembly: Point.js

Syntax

```
JavaScript                                                    Copy Code

get y() {}
set y(value) {}
```

Property Value

Number

The y-coordinate of the point.

See Also

Point Members
Point Class
MindFusion.Drawing Namespace

Rect Class

Represents a rectangle.

Syntax

```
JavaScript                                                    Copy Code

// class
Rect.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.ComponentBase
MindFusion.Drawing.Rect

See Also

Rect Members
MindFusion.Drawing Namespace

Rect Members

Represents a rectangle.

The following tables list the members exposed by the Rect type.

Public Constructors

	Name	Description
	Rect	Overloaded.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	height	Gets or sets the height of the rectangle.

	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)
	width	Gets or sets the width of the rectangle.
	x	Gets or sets the x-coordinate of the upper-left corner of the rectangle.
	y	Gets or sets the y-coordinate of the upper-left corner of the rectangle.

Public Methods

	Name	Description
	bottom	Gets the y-coordinate that is the sum of the y and height values of this **Rect** object.
	bottomLeft	Gets the bottom-left edge of this **Rect** object.
	bottomRight	Gets the bottom-right edge of this **Rect** object.
	center	Gets the center of this **Rect** object.
	clone	Creates a **Rect** object identical to the current object.
	contains	Determines if the rectangular region represented by *rect* is entirely contained within this **Rect** object.
	containsPoint	Determines if the specified point is contained within this **Rect** object.
	equals	Specifies whether this Rect contains the same coordinates as the specified rectangle.
	fromLTRB	Creates a Rect object with the specified edge locations.
	intersect	Returns a **Rect** object representing the intersection of the current rectangle with the specified rectangle.
	intersectsWith	Checks if this **Rect** intersects with the specified rect.
	isEmpty	Gets a value indicating whether this Rect is empty.
	left	Gets the x-coordinate of the left edge of this **Rect** object.
	right	Gets the x-coordinate that is the sum of x and width values of this **Rect**.
	setCenter	Sets the center of this **Rect** object.
	setLocation	Sets the top-left edge of this **Rect** object.
	top	Gets the y-coordinate of the top edge of this **Rect** object.
	topLeft	Gets the top-left edge of this **Rect**.
	topMiddle	Gets the top-middle of this **Rect** object.
	topRight	Gets the top-right edge of this **Rect** object.
	union	Returns a **Rect** object representing the union of the current rectangle with the specified rectangle.

See Also

Rect Class
MindFusion.Drawing Namespace

Rect Constructor

Overload List

Name	Description
Rect (Number | Point, Number | Size, Number, Number)	Initializes a new instance of the Rect class.
Rect (Number, Number, Number, Number)	Initializes a new instance of the Rect class.

See Also

Rect Class
MindFusion.Drawing Namespace

Rect Constructor (Number | Point, Number | Size, Number, Number)

Initializes a new instance of the Rect class.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function Rect (x, y, [width, [height]])
```

Parameters

x
Number | Point. The X-coordinate of the top left corner of the rectangle.

y
Number | Size. The Y-coordinate of the top left corner of the rectangle.

width
Optional.
Number. The width of the rectangle.

height
Optional.
Number. The height of the rectangle.

See Also

Rect Constructor Overload List
Rect Class
MindFusion.Drawing Namespace

Rect Constructor (Number, Number, Number, Number)

Initializes a new instance of the Rect class.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function Rect (x, y, width, height)
```

Parameters

x

The X-coordinate of the top left corner of the rectangle.

y
The Y-coordinate of the top left corner of the rectangle.

width
The width of the rectangle.

height
The height of the rectangle.

See Also

Rect Constructor Overload List
Rect Class
MindFusion.Drawing Namespace

Rect Methods

Rect.bottom Method

Gets the y-coordinate that is the sum of the y and height values of this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function bottom ()
```

Return Value

The y-coordinate that is the sum of y and height of this Rect.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.bottomLeft Method

Gets the bottom-left edge of this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function bottomLeft ()
```

Return Value

The bottom-left edge, which is a Point with x that is equal to x and with y that is the sum of y and height.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.bottomRight Method

Gets the bottom-right edge of this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript — Copy Code

```
function bottomRight ()
```

Return Value

The bottom-right edge, which is a Point with x that is the sum of x and width and with y that is the sum of y and height.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.center Method

Gets the center of this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript — Copy Code

```
function center ()
```

Return Value

The center, which is a Point with x that is the sum of x and half of width and with y that is the sum of y and half of height.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.clone Method

Creates a Rect object identical to the current object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript — Copy Code

```
function clone ()
```

Return Value

Rect. The newly created Rect object.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.contains Method

Determines if the rectangular region represented by *rect* is entirely contained within this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript — Copy Code

```
function contains (rect)
```

Parameters

rect
The Rect to test.

Return Value

true if the rectangular region represented by *rect* is entirely contained within this **Rect**; otherwise, **false**.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.containsPoint Method

Determines if the specified point is contained within this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript — Copy Code

```
function containsPoint (point)
```

Parameters

point
Point. The point to check.

Return Value

true if point is contained within the Rect; otherwise, **false**.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.equals Method

Specifies whether this Rect contains the same coordinates as the specified rectangle.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript — Copy Code

```
function equals (rect)
```

Parameters

rect
Rect. The Rect to test.

Return Value

Boolean. true if rect has the same coordinates as this Rect.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.fromLTRB Method

Creates a Rect object with the specified edge locations.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function fromLTRB (l, t, r, b)
```

Parameters

l
Number. The x-coordinate of the upper-left corner of this Rect.

t
Number. The y-coordinate of the upper-left corner of this Rect.

r
Number. The x-coordinate of the lower-right corner of this Rect.

b
Number. The y-coordinate of the lower-right corner of this Rect.

Return Value

Rect. A rectangle with the specified coordinates.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.intersect Method

Returns a Rect object representing the intersection of the current rectangle with the specified rectangle.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function intersect (rect)
```

Parameters

rect
Rect. The rectangle to intersect.

Return Value

A Rect object representing the intersection of the two rectangles.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.intersectsWith Method

Checks if this Rect intersects with the specified rect.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function intersectsWith (rect)
```

Parameters

rect
The other Rect.

Return Value

true if this **Rect** and the specified **Rect** intersect; otherwise, **false**.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.isEmpty Method

Gets a value indicating whether this Rect is empty.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function isEmpty ()
```

Return Value

true if both width and height are 0; otherwise, **false**.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.left Method

Gets the x-coordinate of the left edge of this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function left ()
```

Return Value

The x-coordinate of the left edge of this Rect object.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.right Method

Gets the x-coordinate that is the sum of x and width values of this Rect.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function right ()
```

Return Value

The x-coordinate that is the sum of x and width of this rectangle.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.setCenter Method

Sets the center of this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function setCenter (point)
```

Parameters

point
The center of this Rect.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.setLocation Method

Sets the top-left edge of this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function setLocation (point)
```

Parameters

point
The top-left edge of the Rect.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.top Method

Gets the y-coordinate of the top edge of this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function top ()
```

Return Value

Number. The y-coordinate of the top edge of this Rect object.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.topLeft Method

Gets the top-left edge of this Rect.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function topLeft ()
```

Return Value

The top-left edge of this Rect.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.topMiddle Method

Gets the top-middle of this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function topMiddle ()
```

Return Value

The top-middle coordinate, which is a Point with x that is the sum of x and half of width and with y equal to y.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.topRight Method

Gets the top-right edge of this Rect object.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function topRight ()
```

Return Value

The top-right edge, which is a Point with x that is the sum of x and width and with y equal to y.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.union Method

Returns a Rect object representing the union of the current rectangle with the specified rectangle.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
function union (rect)
```

Parameters

rect
Rect. The rectangle to union.

Return Value

A Rect object that bounds the union of the two rectangles.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect Properties

Rect.height Property

Gets or sets the height of the rectangle.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
get height() {}
set height(value) {}
```

Property Value

Number

The height of the rectangle.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.width Property

Gets or sets the width of the rectangle.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
get width() {}
set width(value) {}
```

Property Value

Number

The width of the rectangle.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.x Property

Gets or sets the x-coordinate of the upper-left corner of the rectangle.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
get x() {}
set x(value) {}
```

Property Value

Number

The x-coordinate of the upper-left corner of the rectangle.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Rect.y Property

Gets or sets the y-coordinate of the upper-left corner of the rectangle.

Namespace: MindFusion.Drawing
Assembly: Rect.js

Syntax

JavaScript Copy Code

```
get y() {}
set y(value) {}
```

Property Value

Number

The x-coordinate of the upper-left corner of the rectangle.

See Also

Rect Members
Rect Class
MindFusion.Drawing Namespace

Size Class

Represents the size of 2D object.

Syntax

JavaScript Copy Code

```
// class
Size.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.Size

See Also

Size Members
MindFusion.Drawing Namespace

Size Members

Represents the size of 2D object.

The following tables list the members exposed by the Size type.

Public Constructors

	Name	Description
	Size	Initializes a new instance of the **Size** class with the specified width and height.

Public Properties

	Name	Description
	height	Gets or sets the size height.
	width	Gets or sets the size width.

Public Methods

	Name	Description
	empty	Gets a value indicating whether this Size is empty.

See Also

Size Class
MindFusion.Drawing Namespace

Size Constructor

Initializes a new instance of the Size class with the specified width and height.

Namespace: MindFusion.Drawing
Assembly: Size.js

Syntax

JavaScript Copy Code

```
function Size (width, height)
```

Parameters

width
Number. Specifies width.

height
Number. Specifies height.

See Also

Size Class
MindFusion.Drawing Namespace

Size Methods

Size.empty Method

Gets a value indicating whether this Size is empty.

Namespace: MindFusion.Drawing
Assembly: Size.js

Syntax

JavaScript Copy Code

```
function empty ()
```

Return Value

Boolean. true if both width and height are 0; otherwise, false.

See Also

Size Members
Size Class
MindFusion.Drawing Namespace

Size Properties

Size.height Property

Gets or sets the size height.

Namespace: MindFusion.Drawing
Assembly: Size.js

Syntax

JavaScript Copy Code

```
get height() {}
set height(value) {}
```

Property Value

Number

The size height.

See Also

Size Members
Size Class
MindFusion.Drawing Namespace

Size.width Property

Gets or sets the size width.

Namespace: MindFusion.Drawing
Assembly: Size.js

Syntax

JavaScript Copy Code

```
get width() {}
set width(value) {}
```

Property Value

Number

The size width.

See Also

Size Members
Size Class
MindFusion.Drawing Namespace

StringAlignment Enumeration

Specifies the alignment of a text string relative to its layout rectangle.

Namespace: MindFusion.Drawing
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
StringAlignment = {}
```

Members

	Member name	Description
	Near	The text is drawn in the near corner of the layout rectangle.
	Center	The text is drawn in the center of the layout rectangle.
	Far	The text is drawn in the far corner of the layout rectangle.le.

See Also

MindFusion.Drawing Namespace

StringFormat Class

Encapsulates text layout information.

Syntax

JavaScript — Copy Code

```
// class
StringFormat.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.StringFormat

See Also

StringFormat Members
MindFusion.Drawing Namespace

StringFormat Members

Encapsulates text layout information.

The following tables list the members exposed by the StringFormat type.

Public Constructors

	Name	Description
	StringFormat	Initializes a new instance of the StringFormat class.

Public Properties

	Name	Description

	alignment	Gets or sets the horizontal alignment of the text.
	lineAlignment	Gets or sets the vertical alignment of the text.

See Also

StringFormat Class
MindFusion.Drawing Namespace

StringFormat Constructor

Initializes a new instance of the StringFormat class.

Namespace: MindFusion.Drawing
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
function StringFormat ()
```

See Also

StringFormat Class
MindFusion.Drawing Namespace

StringFormat Properties

StringFormat.alignment Property

Gets or sets the horizontal alignment of the text.

Namespace: MindFusion.Drawing
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
get alignment() {}
set alignment(value) {}
```

Property Value

StringAlignment

See Also

StringFormat Members
StringFormat Class
MindFusion.Drawing Namespace

StringFormat.lineAlignment Property

Gets or sets the vertical alignment of the text.

Namespace: MindFusion.Drawing
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
get lineAlignment() {}
set lineAlignment(value) {}
```

Property Value

StringAlignment

See Also

StringFormat Members
StringFormat Class
MindFusion.Drawing Namespace

Text Class

Represents a text container.

Syntax

JavaScript Copy Code

```
// class
Text.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.ComponentBase
MindFusion.Drawing.Text

See Also

Text Members
MindFusion.Drawing Namespace

Text Members

Represents a text container.

The following tables list the members exposed by the Text type.

Public Constructors

	Name	Description
	Text	Initializes a new instance of the **Text** class.

Public Fields

	Name	Description
	bounds	The object's bounds.

Public Properties

	Name	Description
	font	Gets or sets the font used to render text.
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)

	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	lineAlignment	Gets or sets the vertical alignment of this text.
	rotationAngle	Gets a value indicating the rotation of the text container.
	text	Gets or sets the text to render.
	textAlignment	Gets or sets the horizontal alignment of this text.
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

Public Methods

	Name	Description
	clone	Creates a **Text** object identical to the current object.
	getBounds	Gets a value indicating the bounding rectangle of the text container.
	setBounds	Sets a value indicating the bounding rectangle of the text container.

See Also

Text Class
MindFusion.Drawing Namespace

Text Fields

Text.bounds Field

The object's bounds.

Namespace: MindFusion.Drawing
Assembly: Text.js

Syntax

JavaScript Copy Code

```
var bounds
```

See Also

Text Members
Text Class
MindFusion.Drawing Namespace

Text Constructor

Initializes a new instance of the Text class.

Namespace: MindFusion.Drawing
Assembly: Text.js

Syntax

JavaScript Copy Code

```
function Text (text, bounds)
```

Parameters

text
String. The text to be displayed in the container.

bounds
Rect. The bounds of the container.

See Also

Text Class
MindFusion.Drawing Namespace

Text Methods

Text.clone Method

Creates a Text object identical to the current object.

Namespace: MindFusion.Drawing
Assembly: Text.js

Syntax

JavaScript Copy Code

```
function clone ()
```

Return Value

The newly created Text object.

See Also

Text Members
Text Class
MindFusion.Drawing Namespace

Text.getBounds Method

Gets a value indicating the bounding rectangle of the text container.

Namespace: MindFusion.Drawing
Assembly: Text.js

Syntax

JavaScript Copy Code

```
function getBounds ()
```

Return Value

Rect. The bounding rectangle.

See Also

Text Members
Text Class
MindFusion.Drawing Namespace

Text.setBounds Method

Sets a value indicating the bounding rectangle of the text container.

Namespace: MindFusion.Drawing
Assembly: Text.js

Syntax

JavaScript — Copy Code

```
function setBounds (bounds, [angle])
```

Parameters

bounds
Rect. The bounding rectangle.

angle
Optional.
Number. The rotation angle to be applied to the bounding rectangle.

See Also

Text Members
Text Class
MindFusion.Drawing Namespace

Text Properties

Text.font Property

Gets or sets the font used to render text.

Namespace: MindFusion.Drawing
Assembly: Text.js

Syntax

JavaScript — Copy Code

```
get font() {}
set font(value) {}
```

Property Value

Font

The Font of this text.

See Also

Text Members
Text Class
MindFusion.Drawing Namespace

Text.lineAlignment Property

Gets or sets the vertical alignment of this text.

Namespace: MindFusion.Drawing
Assembly: Text.js

Syntax

JavaScript — Copy Code

```
get lineAlignment() {}
set lineAlignment(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

Text Members
Text Class
MindFusion.Drawing Namespace

Text.rotationAngle Property

Gets a value indicating the rotation of the text container.

Namespace: MindFusion.Drawing
Assembly: Text.js

Syntax

JavaScript Copy Code

```
get rotationAngle() {}
set rotationAngle(value) {}
```

Property Value

Number

The angle.

See Also

Text Members
Text Class
MindFusion.Drawing Namespace

Text.text Property

Gets or sets the text to render.

Namespace: MindFusion.Drawing
Assembly: Text.js

Syntax

JavaScript Copy Code

```
get text() {}
set text(value) {}
```

Property Value

String

The text.

See Also

Text Members
Text Class
MindFusion.Drawing Namespace

Text.textAlignment Property

Gets or sets the horizontal alignment of this text.

Namespace: MindFusion.Drawing
Assembly: Text.js

Syntax

JavaScript Copy Code

```
get textAlignment() {}
set textAlignment(value) {}
```

Property Value

Alignment

A member of the Alignment enumeration.

See Also

Text Members
Text Class
MindFusion.Drawing Namespace

Thickness Class

Describes the thickness of a rectangular frame.

Syntax

JavaScript Copy Code

```
// class
Thickness.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.Thickness

See Also

Thickness Members
MindFusion.Drawing Namespace

Thickness Members

Describes the thickness of a rectangular frame.

The following tables list the members exposed by the Thickness type.

Public Constructors

	Name	Description
	Thickness	Overloaded.

Public Properties

	Name	Description
	bottom	Gets or sets the width of the bottom side of the frame.
	isRelative	Gets or sets a flag indicating whether the thickness properties are expressed as relative or absolute quantities.

	left	Gets or sets the width of the left side of the frame.
	right	Gets or sets the width of the right side of the frame.
	top	Gets or sets the width of the top side of the frame.

Public Methods

	Name	Description
	addToRect	Applies the current thickness to the specified rectangle by inflating the rectangle.
	applyTo	Applies the current thickness to the specified rectangle by deflating the rectangle.
	toAbsolute	Converts the current thickness to absolute value relative to the specified size.

See Also

Thickness Class
MindFusion.Drawing Namespace

Thickness Constructor

Overload List

Name	Description
Thickness (Number, Number, Number, Number)	Initializes a new instance of the Thickness class with the specified widths.
Thickness (Number, Number, Number, Number, Boolean)	Initializes a new instance of the Thickness class with the specified widths.

See Also

Thickness Class
MindFusion.Drawing Namespace

Thickness Constructor (Number, Number, Number, Number)

Initializes a new instance of the Thickness class with the specified widths.

Namespace: MindFusion.Drawing
Assembly: Thickness.js

Syntax

JavaScript Copy Code

```
function Thickness (left, top, right, bottom)
```

Parameters

left
Specifies the width of the left side of the frame.

top
Specifies the width of the top side of the frame.

right
Specifies the width of the right side of the frame.

bottom
Specifies the width of the bottom side of the frame.

See Also

Thickness Constructor Overload List
Thickness Class
MindFusion.Drawing Namespace

Thickness Constructor (Number, Number, Number, Number, Boolean)

Initializes a new instance of the Thickness class with the specified widths.

Namespace: MindFusion.Drawing
Assembly: Thickness.js

Syntax

JavaScript Copy Code

```
function Thickness (left, [top, [right, [bottom, [isRelative]]]])
```

Parameters

left
Number. Specifies the width of the left side of the frame.

top
Optional.
Number. Specifies the width of the top side of the frame.

right
Optional.
Number. Specifies the width of the right side of the frame.

bottom
Optional.
Number. Specifies the width of the bottom side of the frame.

isRelative
Optional.
Boolean. A flag indicating whether the thickness properties are expressed as relative or absolute quantities.

See Also

Thickness Constructor Overload List
Thickness Class
MindFusion.Drawing Namespace

Thickness Methods

Thickness.addToRect Method

Applies the current thickness to the specified rectangle by inflating the rectangle.

Namespace: MindFusion.Drawing
Assembly: Thickness.js

Syntax

JavaScript Copy Code

```
function addToRect (rect)
```

Parameters

rect
Rect. The rectangle to inflate.

See Also

Thickness Members
Thickness Class
MindFusion.Drawing Namespace

Thickness.applyTo Method

Applies the current thickness to the specified rectangle by deflating the rectangle.

Namespace: MindFusion.Drawing
Assembly: Thickness.js

Syntax

JavaScript Copy Code

```
function applyTo (rect)
```

Parameters

rect
Rect. The rectangle to deflate.

See Also

Thickness Members
Thickness Class
MindFusion.Drawing Namespace

Thickness.toAbsolute Method

Converts the current thickness to absolute value relative to the specified size.

Namespace: MindFusion.Drawing
Assembly: Thickness.js

Syntax

JavaScript Copy Code

```
function toAbsolute ()
```

See Also

Thickness Members
Thickness Class
MindFusion.Drawing Namespace

Thickness Properties

Thickness.bottom Property

Gets or sets the width of the bottom side of the frame.

Namespace: MindFusion.Drawing
Assembly: Thickness.js

Syntax

JavaScript Copy Code

```
get bottom() {}
set bottom(value) {}
```

Property Value

Number

The width of the bottom side of the frame.

See Also

Thickness Members
Thickness Class
MindFusion.Drawing Namespace

Thickness.isRelative Property

Gets or sets a flag indicating whether the thickness properties are expressed as relative or absolute quantities.

Namespace: MindFusion.Drawing
Assembly: Thickness.js

Syntax

JavaScript Copy Code

```
get isRelative() {}
set isRelative(value) {}
```

Property Value

Boolean

true if the thickness is relative, or **false** otherwise.

See Also

Thickness Members
Thickness Class
MindFusion.Drawing Namespace

Thickness.left Property

Gets or sets the width of the left side of the frame.

Namespace: MindFusion.Drawing
Assembly: Thickness.js

Syntax

JavaScript Copy Code

```
get left() {}
set left(value) {}
```

Property Value

Number

The width of the left side of the frame.

See Also

Thickness Members
Thickness Class
MindFusion.Drawing Namespace

Thickness.right Property

Gets or sets the width of the right side of the frame.

Namespace: MindFusion.Drawing
Assembly: Thickness.js

Syntax

JavaScript Copy Code

```
get right() {}
set right(value) {}
```

Property Value

Number

The width of the right side of the frame.

See Also

Thickness Members
Thickness Class
MindFusion.Drawing Namespace

Thickness.top Property

Gets or sets the width of the top side of the frame.

Namespace: MindFusion.Drawing
Assembly: Thickness.js

Syntax

JavaScript Copy Code

```
get top() {}
set top(value) {}
```

Property Value

Number

The width of the top side of the frame.

See Also

Thickness Members
Thickness Class
MindFusion.Drawing Namespace

Vector Class

Represents displacement in 2D space.

Syntax

JavaScript Copy Code

```
// class
Vector.prototype = {}
```

Inheritance Hierarchy

MindFusion.Drawing.Vector

See Also

Vector Members
MindFusion.Drawing Namespace

Vector Members

Represents displacement in 2D space.

The following tables list the members exposed by the Vector type.

Public Constructors

	Name	Description
	Vector	Initializes a new instance of the Vector class with the specified X and Y coordinates.

Public Fields

	Name	Description
	x	The x-coordinate of the vector.
	y	The y-coordinate of the vector.

See Also

Vector Class
MindFusion.Drawing Namespace

Vector Fields

Vector.x Field

The x-coordinate of the vector.

Namespace: MindFusion.Drawing
Assembly: Vector.js

Syntax

JavaScript Copy Code

```
var x
```

See Also

Vector Members
Vector Class
MindFusion.Drawing Namespace

Vector.y Field

The y-coordinate of the vector.

Namespace: MindFusion.Drawing
Assembly: Vector.js

Syntax

JavaScript Copy Code

```
var y
```

See Also

Vector Members
Vector Class
MindFusion.Drawing Namespace

Vector Constructor

Initializes a new instance of the Vector class with the specified X and Y coordinates.

Namespace: MindFusion.Drawing
Assembly: Vector.js

Syntax

JavaScript Copy Code

```
function Vector (x, y)
```

Parameters

x
Number. The X-coordinate of the Vector.

y
Number. The Y-coordinate of the Vector.

See Also

Vector Class
MindFusion.Drawing Namespace

Video Class

A component that displays video stream.

Syntax

JavaScript Copy Code

```
// class
Video.prototype = {}
```

Remarks

The video does not play automatically. To start playing it, call the component's play method in response to user input events.

If autoProperty is enabled for a **Video** component in CompositeNode template, the template loader creates getter and setter wrapping the component's MediaLocation property.

Example

The following template fragment adds a Video component to the node:

JavaScript Copy Code

```
{
    component: "Video",
    name: "Video",
    autoProperty: true,
    mediaLocation: "video.webm",
    margin: "1"
},
```

It automatically adds getVideo and setVideo methods to the node class, which can be used to get or set the video stream URL:

JavaScript Copy Code

```
var VideoNode = CompositeNode.classFromTemplate(
    "VideoNode", templateJson);
var node = new VideoNode();
node.setVideo("video2.webm");
diagram.addItem(node);
```

Inheritance Hierarchy

MindFusion.Drawing.ComponentBase
  MindFusion.Drawing.Video

See Also

Video Members
MindFusion.Drawing Namespace

Video Members

A component that displays video stream.

The following tables list the members exposed by the Video type.

Public Constructors

	Name	Description
	Video	Initializes a new instance of the Video class.

Public Fields

	Name	Description
	bounds	The object's bounds.

Public Properties

	Name	Description
	gridColumn	Gets or sets the index of the grid column this component is contained in. (Inherited from ComponentBase.)
	gridRow	Gets or sets the index of the grid row this component is contained in. (Inherited from ComponentBase.)
	horizontalAlignment	Gets or sets the component's horizontal alignment. (Inherited from ComponentBase.)
	mediaLocation	Gets or sets URL of the video stream displayed in this component.
	verticalAlignment	Gets or sets the component's vertical alignment. (Inherited from ComponentBase.)
	visibility	Gets or sets the visibility of this component. (Inherited from ComponentBase.)

Public Methods

	Name	Description
	clone	Creates a Video object identical to the current object.
	isPlaying	Detects if video is playing.
	pause	Pauses the video.

	play	Starts playing the video stream.

See Also

Video Class
MindFusion.Drawing Namespace

Video Fields

Video.bounds Field

The object's bounds.

Namespace: MindFusion.Drawing
Assembly: Video.js

Syntax

JavaScript Copy Code

```
var bounds
```

See Also

Video Members
Video Class
MindFusion.Drawing Namespace

Video Constructor

Initializes a new instance of the Video class.

Namespace: MindFusion.Drawing
Assembly: Video.js

Syntax

JavaScript Copy Code

```
function Video (bounds)
```

Parameters

bounds
Rect. The bounds of the Video.

See Also

Video Class
MindFusion.Drawing Namespace

Video Methods

Video.clone Method

Creates a Video object identical to the current object.

Namespace: MindFusion.Drawing
Assembly: Video.js

Syntax

JavaScript Copy Code

```
function clone ()
```

Return Value

See Also

Video Members
Video Class
MindFusion.Drawing Namespace

Video.isPlaying Method

Detects if video is playing.

Namespace: MindFusion.Drawing
Assembly: Video.js

Syntax

JavaScript Copy Code

```
function isPlaying ()
```

See Also

Video Members
Video Class
MindFusion.Drawing Namespace

Video.pause Method

Pauses the video.

Namespace: MindFusion.Drawing
Assembly: Video.js

Syntax

JavaScript Copy Code

```
function pause ()
```

See Also

Video Members
Video Class
MindFusion.Drawing Namespace

Video.play Method

Starts playing the video stream.

Namespace: MindFusion.Drawing
Assembly: Video.js

Syntax

JavaScript Copy Code

```
function play ()
```

See Also

Video Members
Video Class
MindFusion.Drawing Namespace

Video Properties

Video.mediaLocation Property

Gets or sets URL of the video stream displayed in this component.

Namespace: MindFusion.Drawing
Assembly: Video.js

Syntax

JavaScript Copy Code

```
get mediaLocation() {}
set mediaLocation(value) {}
```

Property Value

String

A URL string that identifies the video to play.

See Also

Video Members
Video Class
MindFusion.Drawing Namespace

Visibility Enumeration

Specifies visibility for components.

Namespace: MindFusion.Drawing
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
Visibility = {}
```

Members

	Member name	Description
	Hidden	Specifies that the component is hidden, but the space it occupies in the layout is preserved.
	Collapsed	Specifies that the component is hidden, and the space it occupies in the layout is not preserved.
	Visible	Specifies that the component is visible.

See Also

MindFusion.Graphs Namespace

Anchoring Enumeration

Defines values that specify how automatic layout algorithms align links to anchor points.

Namespace: MindFusion.Graphs
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
Anchoring = {}
```

Members

	Member name	Description
	Ignore	Anchor points are ignored.
	Keep	Links are aligned to their original anchor points.
	Reassign	Links are aligned to new anchor points, depending on the positions of graph nodes after a layout is applied.
	Custom	Custom alignment logic.

Remarks

Members of this enumeration can be assigned to the anchoring property of layout classes.

See Also

MindFusion.Graphs Namespace

AssistantPosition Enumeration

Specifies the final position of an assistant node. Assistants with type left or right are always positioned to the left or right respectively. The position of the "normal" assistants depends on their index in the children list.

Namespace: MindFusion.Graphs
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
AssistantPosition = {}
```

Members

	Member name	Description
	Left	The node is distributed to the left of its parent.
	Right	The node is distributed to the right of its parent.

See Also

MindFusion.Graphs Namespace

AssistantType Enumeration

Indicates the type of an assistant node in a tree.

Namespace: MindFusion.Graphs
Assembly: Enum.js

Syntax

JavaScript — Copy Code

```
// enum
AssistantType = {}
```

Members

	Member name	Description
	Normal	The node is an assistant and its relative position is determinedautomatically by the layout.
	Left	The node is an assistant and it is always positioned to the left of its parent.
	Right	The node is an assistant and it is always positioned to the right of its parent.

Remarks

Members of this enumeration can be assigned to the **treeLayoutAssistant** key of node's layoutTraits.

See Also

MindFusion.Graphs Namespace

BorderedTreeLayout Class

Implements algorithms for arranging tree structures.

Syntax

JavaScript — Copy Code

```
// class
BorderedTreeLayout.prototype = {}
```

Remarks

To apply this layout to a diagram, create a **BorderedTreeLayout** instance, set its members and invoke the arrange method of the Diagram class. The method lays out all subtrees in a diagram.

As long as the graph is a tree, the algorithm can find the root node automatically. The distance between the root and the second level and between all subsequent levels can be set through the levelDistance property. The distance between adjacent nodes in the same level is set through nodeDistance.

In addition to arranging the tree nodes, **BorderedTreeLayout** also sets the shape of links as set through its linkType property.

Inheritance Hierarchy

MindFusion.Graphs.Layout
  MindFusion.Graphs.BorderedTreeLayout

See Also

BorderedTreeLayout Members

Implements algorithms for arranging tree structures.

The following tables list the members exposed by the BorderedTreeLayout type.

Public Constructors

	Name	Description
	BorderedTreeLayout	Initializes a new instance of the BorderedTreeLayout class.

Public Properties

	Name	Description
	anchoring	Gets or sets a value indicating how to align links to the anchor points of nodes.
	direction	Gets or sets the orientation of the arranged graph.
	keepGroupLayout	Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.
	keepRootPosition	Gets or sets a value indicating whether to keep the root node at its original position.
	levelDistance	Gets or sets the space to leave between adjacent levels of the tree.
	linkType	Gets or sets the type of links in the arranged tree.
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object. (Inherited from Layout.)
	multipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.
	nodeDistance	Gets or sets the space to leave between adjacent nodes on the same level.
	reversedLinks	Gets or sets a value indicating the link direction that relates parent nodes to child nodes.

Public Methods

	Name	Description
	arrange	Applies the layout to the specified graph.

See Also

BorderedTreeLayout Class
MindFusion.Graphs Namespace

BorderedTreeLayout Constructor

Initializes a new instance of the BorderedTreeLayout class.

Namespace: MindFusion.Graphs
Assembly: BorderedTreeLayout.js

Syntax

JavaScript Copy Code

```
function BorderedTreeLayout ()
```

See Also

BorderedTreeLayout Class
MindFusion.Graphs Namespace

BorderedTreeLayout Methods

BorderedTreeLayout.arrange Method

Applies the layout to the specified graph.

Namespace: MindFusion.Graphs
Assembly: BorderedTreeLayout.js

Syntax

JavaScript Copy Code

```
function arrange (graph)
```

Parameters

graph
The Graph to arrange.

Remarks

This method is called internally by Diagram.arrange and ContainerNode.arrange.

See Also

BorderedTreeLayout Members
BorderedTreeLayout Class
MindFusion.Graphs Namespace

BorderedTreeLayout Properties

BorderedTreeLayout.anchoring Property

Gets or sets a value indicating how to align links to the anchor points of nodes.

Namespace: MindFusion.Graphs
Assembly: BorderedTreeLayout.js

Syntax

JavaScript Copy Code

```
get anchoring() {}
set anchoring(value) {}
```

Property Value

Anchoring

A member of the Anchoring enumeration. The default value is **Ignore**.

See Also

BorderedTreeLayout Members
BorderedTreeLayout Class
MindFusion.Graphs Namespace

BorderedTreeLayout.direction Property

Gets or sets the orientation of the arranged graph.

Namespace: MindFusion.Graphs
Assembly: BorderedTreeLayout.js

Syntax

JavaScript Copy Code

```
get direction() {}
set direction(value) {}
```

Property Value

LayoutDirection

A member of the LayoutDirection enumeration. The default value is **TopToBottom**.

See Also

BorderedTreeLayout Members
BorderedTreeLayout Class
MindFusion.Graphs Namespace

BorderedTreeLayout.keepGroupLayout Property

Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.

Namespace: MindFusion.Graphs
Assembly: BorderedTreeLayout.js

Syntax

JavaScript Copy Code

```
get keepGroupLayout() {}
set keepGroupLayout(value) {}
```

Property Value

Boolean

true if each group of attached nodes should be treated as a single vertex in the arranged graph, or **false** otherwise.

See Also

BorderedTreeLayout Members
BorderedTreeLayout Class
MindFusion.Graphs Namespace

BorderedTreeLayout.keepRootPosition Property

Gets or sets a value indicating whether to keep the root node at its original position.

Namespace: MindFusion.Graphs
Assembly: BorderedTreeLayout.js

Syntax

JavaScript Copy Code

```
get keepRootPosition() {}
set keepRootPosition(value) {}
```

Property Value

Boolean

true if the root node should be kept at its original position, or **false** otherwise.

See Also

BorderedTreeLayout Members
BorderedTreeLayout Class
MindFusion.Graphs Namespace

BorderedTreeLayout.levelDistance Property

Gets or sets the space to leave between adjacent levels of the tree.

Namespace: MindFusion.Graphs
Assembly: BorderedTreeLayout.js

Syntax

JavaScript Copy Code

```
get levelDistance() {}
set levelDistance(value) {}
```

Property Value

Number

The space to leave between adjacent levels of the tree.

See Also

BorderedTreeLayout Members
BorderedTreeLayout Class
MindFusion.Graphs Namespace

BorderedTreeLayout.linkType Property

Gets or sets the type of links in the arranged tree.

Namespace: MindFusion.Graphs
Assembly: BorderedTreeLayout.js

Syntax

JavaScript Copy Code

```
get linkType() {}
set linkType(value) {}
```

Property Value

TreeLayoutLinkType

A member of the TreeLayoutLinkType enumeration. The default value is **Default**,

See Also

BorderedTreeLayout Members
BorderedTreeLayout Class
MindFusion.Graphs Namespace

BorderedTreeLayout.multipleGraphsPlacement Property

Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.

Namespace: MindFusion.Graphs
Assembly: BorderedTreeLayout.js

Syntax

JavaScript Copy Code

```
get multipleGraphsPlacement() {}
set multipleGraphsPlacement(value) {}
```

Property Value

MultipleGraphsPlacement

A member of the MultipleGraphsPlacement enumeration. The default value is **Horizontal**.

See Also

BorderedTreeLayout Members
BorderedTreeLayout Class
MindFusion.Graphs Namespace

BorderedTreeLayout.nodeDistance Property

Gets or sets the space to leave between adjacent nodes on the same level.

Namespace: MindFusion.Graphs
Assembly: BorderedTreeLayout.js

Syntax

JavaScript Copy Code

```
get nodeDistance() {}
set nodeDistance(value) {}
```

Property Value

Number

The space to leave between adjacent nodes on the same level.

See Also

BorderedTreeLayout Members
BorderedTreeLayout Class
MindFusion.Graphs Namespace

BorderedTreeLayout.reversedLinks Property

Gets or sets a value indicating the link direction that relates parent nodes to child nodes.

Namespace: MindFusion.Graphs
Assembly: BorderedTreeLayout.js

Syntax

JavaScript Copy Code

```
get reversedLinks() {}
set reversedLinks(value) {}
```

Property Value

Boolean

true to indicate reversed direction (links point to parent nodes), or **false** otherwise.

See Also

BorderedTreeLayout Members
BorderedTreeLayout Class
MindFusion.Graphs Namespace

Edge Class

Represents an edge connecting two vertices in a graph.

Syntax

JavaScript Copy Code

```
// class
Edge.prototype = {}
```

Inheritance Hierarchy

MindFusion.Graphs.Edge

See Also

Edge Members
MindFusion.Graphs Namespace

Edge Members

Represents an edge connecting two vertices in a graph.

The following tables list the members exposed by the Edge type.

Public Constructors

	Name	Description
	Edge	Initializes a new instance of the **Edge** class.

Public Fields

	Name	Description
	destination	The destination vertex.
	origin	The origin vertex.

Public Methods

	Name	Description
	adjacentTo	Checks if the current edge is adjacent to the specified edge, that is, they have common vertex.
	changeDestination	Changes the destination of the current edge to the specified vertex.
	changeOrigin	Changes the origin of the current edge to the specified vertex.
	createReverseEdge	Creates a new edge similar to the current edge but with reversed direction.
	getCommonVertex	Returns a common vertex for the current edge and the specified edge.
	getEnds	Returns an array containing the origin and destination vertices of this edge.

	getOtherEnd	Returns the vertex connected by this edge that is not the specified vertex.
	incidentWith	Checks if the current edge is incident with the specified vertex.
	joins	Checks if the current edge connects the specified vertices.
	reverse	Reverses this edge.

See Also

Edge Class
MindFusion.Graphs Namespace

Edge Fields

Edge.destination Field

The destination vertex.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
var destination
```

See Also

Edge Members
Edge Class
MindFusion.Graphs Namespace

Edge.origin Field

The origin vertex.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
var origin
```

See Also

Edge Members
Edge Class
MindFusion.Graphs Namespace

Edge Constructor

Initializes a new instance of the Edge class.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
function Edge (origin, destination)
```

Parameters

origin
Vertex. The origin vertex.

destination
Vertex. The destination vertex.

See Also

Edge Class
MindFusion.Graphs Namespace

Edge Methods

Edge.adjacentTo Method

Checks if the current edge is adjacent to the specified edge, that is, they have common vertex.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
function adjacentTo (edge)
```

Parameters

edge
Edge. The edge to check.

Return Value

true if the current edge is adjacent to the specified edge; otherwise, **false**.

See Also

Edge Members
Edge Class
MindFusion.Graphs Namespace

Edge.changeDestination Method

Changes the destination of the current edge to the specified vertex.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
function changeDestination (vertex)
```

Parameters

vertex
Vertex. The new destination of the edge.

See Also

Edge Members
Edge Class
MindFusion.Graphs Namespace

Edge.changeOrigin Method

Changes the origin of the current edge to the specified vertex.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
function changeOrigin (vertex)
```

Parameters

vertex
Vertex. The new origin of the edge.

See Also

Edge Members
Edge Class
MindFusion.Graphs Namespace

Edge.createReverseEdge Method

Creates a new edge similar to the current edge but with reversed direction.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
function createReverseEdge ()
```

Return Value

See Also

Edge Members
Edge Class
MindFusion.Graphs Namespace

Edge.getCommonVertex Method

Returns a common vertex for the current edge and the specified edge.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
function getCommonVertex (edge)
```

Parameters

edge

Edge. The edge for which to find common vertex.

Return Value

The common vertex if it exists, otherwise **null**.

See Also

Edge Members
Edge Class
MindFusion.Graphs Namespace

Edge.getEnds Method

Returns an array containing the origin and destination vertices of this edge.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
function getEnds ()
```

Return Value

Array. An array containing the related vertices.

See Also

Edge Members
Edge Class
MindFusion.Graphs Namespace

Edge.getOtherEnd Method

Returns the vertex connected by this edge that is not the specified vertex.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
function getOtherEnd (vertex)
```

Parameters

vertex
Vertex. One of the vertices connected by this edge.

Return Value

Vertex. The other vertex connected by this edge.

See Also

Edge Members
Edge Class
MindFusion.Graphs Namespace

Edge.incidentWith Method

Checks if the current edge is incident with the specified vertex.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
function incidentWith (vertex)
```

Parameters

vertex
Vertex. The vertex to check.

Return Value

true if the edge is incident with the vertex; otherwise, **false**.

See Also

Edge Members
Edge Class
MindFusion.Graphs Namespace

Edge.joins Method

Checks if the current edge connects the specified vertices.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
function joins (v1, v2)
```

Parameters

v1
Vertex. The first vertex.

v2
Vertex. The second vertex.

Return Value

true if the current edge connects the specified vertices; otherwise, **false**.

See Also

Edge Members
Edge Class
MindFusion.Graphs Namespace

Edge.reverse Method

Reverses this edge.

Namespace: MindFusion.Graphs
Assembly: Edge.js

Syntax

JavaScript Copy Code

```
function reverse ()
```

See Also

FlowchartLayout Class

FlowchartLayout can be used to arrange flowcharts representing program source code.

Syntax

JavaScript Copy Code

```
// class
FlowchartLayout.prototype = {}
```

Remarks

The algorithm recognizes program code -like patterns in the graph, such as loops, sequences and if/switch branchings, and arranges them recursively. **FlowchartLayout** could be used to arrange other types of graphs as well, though with some restrictions. For example it treats all back links as loops in the code, and expects that they are nested - loop links starting closer to the stop node should end closer to the start node. Another similar restriction is that there shouldn't be any cross-links that connect different branches of a decision sub-graph.

Inheritance Hierarchy

MindFusion.Graphs.Layout
  MindFusion.Graphs.FlowchartLayout

See Also

FlowchartLayout Members
MindFusion.Graphs Namespace

FlowchartLayout Members

FlowchartLayout can be used to arrange flowcharts representing program source code.

The following tables list the members exposed by the **FlowchartLayout** type.

Public Constructors

	Name	Description
	FlowchartLayout	Initializes a new instance of the **FlowchartLayout** class.

Public Properties

	Name	Description
	anchoring	Gets or sets a value indicating how to align links to the anchor points of nodes.
	branchPadding	Gets or sets how much space to leave between adjacent decision branches in the flowchart.
	keepGroupLayout	Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.
	linkPadding	Gets or sets how much space to leave between adjacent back links designating nested loops.
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object. (Inherited from Layout.)
	multipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.

	nodeDistance	Gets or sets the distance between consecutive nodes in the flowchart.
	orientation	Gets or sets the orientation of the arranged graph.

Public Methods

	Name	Description
	init	Returns a Builder object used to configure this **FlowchartLayout** instance.
S	With	Returns a Builder object used to configure and create new **FlowchartLayout** instances.

See Also

FlowchartLayout Class
MindFusion.Graphs Namespace

FlowchartLayout Constructor

Initializes a new instance of the FlowchartLayout class.

Namespace: MindFusion.Graphs
Assembly: FlowchartLayout.js

Syntax

JavaScript Copy Code

```
function FlowchartLayout ()
```

See Also

FlowchartLayout Class
MindFusion.Graphs Namespace

FlowchartLayout Methods

FlowchartLayout.init Method

Returns a Builder object used to configure this FlowchartLayout instance.

Namespace: MindFusion.Graphs
Assembly: FlowchartLayout.js

Syntax

JavaScript Copy Code

```
function init ()
```

Return Value

A Builder instance.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Graphs Namespace

FlowchartLayout.With Method

Returns a Builder object used to configure and create new FlowchartLayout instances.

Namespace: MindFusion.Graphs
Assembly: FlowchartLayout.js

Syntax

JavaScript Copy Code

```
function With ()
```

Return Value

A Builder instance.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Graphs Namespace

FlowchartLayout Properties

FlowchartLayout.anchoring Property

Gets or sets a value indicating how to align links to the anchor points of nodes.

Namespace: MindFusion.Graphs
Assembly: FlowchartLayout.js

Syntax

JavaScript Copy Code

```
get anchoring() {}
set anchoring(value) {}
```

Property Value

Anchoring

A member of the Anchoring enumeration.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Graphs Namespace

FlowchartLayout.branchPadding Property

Gets or sets how much space to leave between adjacent decision branches in the flowchart.

Namespace: MindFusion.Graphs
Assembly: FlowchartLayout.js

Syntax

JavaScript Copy Code

```
get branchPadding() {}
set branchPadding(value) {}
```

Property Value

Number

A numeric value specifying the distance between adjacent branches.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Graphs Namespace

FlowchartLayout.keepGroupLayout Property

Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.

Namespace: MindFusion.Graphs
Assembly: FlowchartLayout.js

Syntax

JavaScript Copy Code

```
get keepGroupLayout() {}
set keepGroupLayout(value) {}
```

Property Value

Boolean

true if each group of attached nodes should be treated as a single vertex in the arranged graph, or **false** otherwise.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Graphs Namespace

FlowchartLayout.linkPadding Property

Gets or sets how much space to leave between adjacent back links designating nested loops.

Namespace: MindFusion.Graphs
Assembly: FlowchartLayout.js

Syntax

JavaScript Copy Code

```
get linkPadding() {}
set linkPadding(value) {}
```

Property Value

Number

A numeric value specifying the distance between adjacent back links.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Graphs Namespace

FlowchartLayout.multipleGraphsPlacement Property

Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.

Namespace: MindFusion.Graphs
Assembly: FlowchartLayout.js

Syntax

JavaScript Copy Code

```
get multipleGraphsPlacement() {}
set multipleGraphsPlacement(value) {}
```

Property Value

MultipleGraphsPlacement

A member of the MultipleGraphsPlacement enumeration.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Graphs Namespace

FlowchartLayout.nodeDistance Property

Gets or sets the distance between consecutive nodes in the flowchart.

Namespace: MindFusion.Graphs
Assembly: FlowchartLayout.js

Syntax

JavaScript Copy Code

```
get nodeDistance() {}
set nodeDistance(value) {}
```

Property Value

Number

A numeric value specifying the distance between neighbor nodes in the same sequence.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Graphs Namespace

FlowchartLayout.orientation Property

Gets or sets the orientation of the arranged graph.

Namespace: MindFusion.Graphs
Assembly: FlowchartLayout.js

Syntax

JavaScript Copy Code

```
get orientation() {}
set orientation(value) {}
```

Property Value

Orientation

A member of the Orientation enumeration.

See Also

FractalLayout Class

The FractalLayout tree layout algorithm places child nodes symmetrically around their parent node. Nodes at the lowest level are arranged directly in a circle around their parent. At the upper level, the already arranged nodes form branches that are arranged in a circle around the new parent node. The algorithm is recursively repeated till the highest level is reached.

Syntax

JavaScript Copy Code

```
// class
FractalLayout.prototype = {}
```

Inheritance Hierarchy

MindFusion.Graphs.Layout
  MindFusion.Graphs.FractalLayout

See Also

FractalLayout Members
MindFusion.Graphs Namespace

FractalLayout Members

The FractalLayout tree layout algorithm places child nodes symmetrically around their parent node. Nodes at the lowest level are arranged directly in a circle around their parent. At the upper level, the already arranged nodes form branches that are arranged in a circle around the new parent node. The algorithm is recursively repeated till the highest level is reached.

The following tables list the members exposed by the FractalLayout type.

Public Constructors

	Name	Description
	FractalLayout	Initializes a new instance of the FractalLayout class.

Public Properties

	Name	Description
	anchoring	Gets or sets a value indicating how to align links to the anchor points of nodes.
	keepGroupLayout	Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object. (Inherited from Layout.)
	multipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.
	root	Gets or sets the node that should be placed at the center of the tree.

See Also

FractalLayout Class
MindFusion.Graphs Namespace

FractalLayout Fields

FractalLayout Constructor

Initializes a new instance of the FractalLayout class.

Namespace: MindFusion.Graphs
Assembly: FractalLayout.js

Syntax

JavaScript Copy Code

```
function FractalLayout ()
```

See Also

FractalLayout Class
MindFusion.Graphs Namespace

FractalLayout Properties

FractalLayout.anchoring Property

Gets or sets a value indicating how to align links to the anchor points of nodes.

Namespace: MindFusion.Graphs
Assembly: FractalLayout.js

Syntax

JavaScript Copy Code

```
get anchoring() {}
set anchoring(value) {}
```

Property Value

Anchoring

A member of the Anchoring enumeration. The default value is **Ignore**.

See Also

FractalLayout Members
FractalLayout Class
MindFusion.Graphs Namespace

FractalLayout.keepGroupLayout Property

Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.

Namespace: MindFusion.Graphs
Assembly: FractalLayout.js

Syntax

JavaScript Copy Code

```
get keepGroupLayout() {}
set keepGroupLayout(value) {}
```

Property Value

Boolean

true if each group of attached nodes should be treated as a single vertex in the arranged graph, or **false** otherwise.

See Also

FractalLayout Members
FractalLayout Class
MindFusion.Graphs Namespace

FractalLayout.multipleGraphsPlacement Property

Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.

Namespace: MindFusion.Graphs
Assembly: FractalLayout.js

Syntax

JavaScript Copy Code

```
get multipleGraphsPlacement() {}
set multipleGraphsPlacement(value) {}
```

Property Value

MultipleGraphsPlacement

A member of the MultipleGraphsPlacement enumeration. The default value is **Horizontal**.

See Also

FractalLayout Members
FractalLayout Class
MindFusion.Graphs Namespace

FractalLayout.root Property

Gets or sets the node that should be placed at the center of the tree.

Namespace: MindFusion.Graphs
Assembly: FractalLayout.js

Syntax

JavaScript Copy Code

```
get root() {}
set root(value) {}
```

Property Value

DiagramNode

The node that should be placed at the center of the tree.

See Also

FractalLayout Members
FractalLayout Class
MindFusion.Graphs Namespace

Graph Class

Represents a graph.

Syntax

JavaScript Copy Code

```
// class
Graph.prototype = {}
```

Inheritance Hierarchy

MindFusion.Graphs.Graph

See Also

Graph Members
MindFusion.Graphs Namespace

Graph Members

Represents a graph.

The following tables list the members exposed by the Graph type.

Public Constructors

	Name	Description
	Graph	Initializes a new instance of the **Graph** class.

Public Fields

	Name	Description
	edges	A list containing all edges in the graph.
	vertices	A list containing all vertices in the graph.

Public Methods

	Name	Description
	addEdge	Adds an existing edge to the graph.
	clone	Creates an exact copy of the current graph.
	createEdge	Creates a new edge connecting the specified vertices and adds it to the graph.
	createVertex	Creates a new vertex with the specified position and size and adds it to the graph.
	getConnectedComponents	Returns a list with the connected subgraphs in the current graph.
	getUpwardEmbedding	Should be called on a planar graph.
	makeAcyclic	Makes the graph acyclic by reversing the direction of selected edges.
	removeEdge	Removes an edge from the graph.
	tree	Overloaded.

See Also

Graph Class
MindFusion.Graphs Namespace

Graph Fields

Graph.edges Field

A list containing all edges in the graph.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript — Copy Code

```
var edges
```

See Also

Graph Members
Graph Class
MindFusion.Graphs Namespace

Graph.vertices Field

A list containing all vertices in the graph.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript — Copy Code

```
var vertices
```

See Also

Graph Members
Graph Class
MindFusion.Graphs Namespace

Graph Constructor

Initializes a new instance of the Graph class.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript — Copy Code

```
function Graph ()
```

See Also

Graph Class
MindFusion.Graphs Namespace

Graph Methods

Graph.addEdge Method

Adds an existing edge to the graph.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript — Copy Code

```
function addEdge (edge)
```

Parameters

edge
Edge. The edge to add.

See Also

Graph Members
Graph Class
MindFusion.Graphs Namespace

Graph.clone Method

Creates an exact copy of the current graph.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript — Copy Code

```
function clone ([saveMapping])
```

Parameters

saveMapping
Optional.
Boolean. Indicates whether to store vertex and edge mapping information in the copy.

Return Value

Graph. The newly created copy.

See Also

Graph Members
Graph Class
MindFusion.Graphs Namespace

Graph.createEdge Method

Creates a new edge connecting the specified vertices and adds it to the graph.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript — Copy Code

```
function createEdge (origin, destination, [owner])
```

Parameters

origin

Vertex. The origin vertex.

destination
Vertex. The destination vertex.

owner
Optional.
Object. The owner of the new edge.

Return Value

Edge. The newly created edge.

See Also

Graph Members
Graph Class
MindFusion.Graphs Namespace

Graph.createVertex Method

Creates a new vertex with the specified position and size and adds it to the graph.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript Copy Code

```
function createVertex (layoutRect, [owner])
```

Parameters

layoutRect
Rect. A rectangle specifying the position and size of the vertex.

owner
Optional.
Object. The owner of the new vertex.

Return Value

Vertex. The newly created vertex.

See Also

Graph Members
Graph Class
MindFusion.Graphs Namespace

Graph.getConnectedComponents Method

Returns a list with the connected subgraphs in the current graph.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript Copy Code

```
function getConnectedComponents ()
```

Return Value

An array with connected Graph objects.

See Also

Graph Members
Graph Class

Graph.getUpwardEmbedding Method

Should be called on a planar graph.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript Copy Code

```
function getUpwardEmbedding ([embedding, [embeddingInfo]])
```

Parameters

embedding
Optional.
Map. Receives the embedding lists.

embeddingInfo
Optional.
Map. Optional.

Return Value

Vertex. The t node in st ordering.

See Also

Graph Members
Graph Class
MindFusion.Graphs Namespace

Graph.makeAcyclic Method

Makes the graph acyclic by reversing the direction of selected edges.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript Copy Code

```
function makeAcyclic ()
```

Return Value

Array. A list with all edges that were reversed in order to make the graph acyclic.

Remarks

Must be called on a simple connected graph.

See Also

Graph Members
Graph Class
MindFusion.Graphs Namespace

Graph.removeEdge Method

Removes an edge from the graph.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript — Copy Code

```
function removeEdge (edge)
```

Parameters

edge
Edge. The edge to remove.

See Also

Graph Members
Graph Class
MindFusion.Graphs Namespace

Graph.tree Method

Overload List

Name	Description
Graph.tree (Vertex)	Returns a new graph object containing a subset of the vertices and edges of the original graph, such that the new graph is a tree.
Graph.tree (Vertex, Boolean, Boolean)	Returns a new graph object containing a subset of the vertices and edges of the original graph, such that the new graph is a tree.

See Also

Graph Members
Graph Class
MindFusion.Graphs Namespace

Graph.tree Method (Vertex)

Returns a new graph object containing a subset of the vertices and edges of the original graph, such that the new graph is a tree.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript — Copy Code

```
function tree (root)
```

Parameters

root
The desired root vertex of the tree.

Return Value

The newly created tree.

See Also

tree Method Overload List
Graph Members
Graph Class
MindFusion.Graphs Namespace

Graph.tree Method (Vertex, Boolean, Boolean)

Returns a new graph object containing a subset of the vertices and edges of the original graph, such that the new graph is a tree.

Namespace: MindFusion.Graphs
Assembly: Graph.js

Syntax

JavaScript Copy Code

```
function tree (root, enableAssistants, compactAssistants)
```

Parameters

root
Vertex. The desired root vertex of the tree.

enableAssistants

compactAssistants

Return Value

Graph. The newly created tree.

See Also

tree Method Overload List
Graph Members
Graph Class
MindFusion.Graphs Namespace

GridLayout Class

Implements a grid graph layout algorithm.

Syntax

JavaScript Copy Code

```
// class
GridLayout.prototype = {}
```

Remarks

GridLayout arranges diagram nodes in a grid, keeping connected nodes close to each other. The algorithm strives to achieve a small number of link crossings. It is based on an iterative process whose initial steps shuffle the grid nodes randomly. That can lead to very different results each time the algorithm is run.

To apply the layout to a diagram, create a **GridLayout** instance, set its property values, and invoke the arrange method.

The distance between points in the layout grid can be changed through gridSize. Nodes to be placed at the upper-left and lower-right corners of the grid can be assigned to startNode and endNode.

Inheritance Hierarchy

MindFusion.Graphs.Layout
  MindFusion.Graphs.GridLayout

See Also

GridLayout Members
MindFusion.Graphs Namespace

GridLayout Members

Implements a grid graph layout algorithm.

The following tables list the members exposed by the GridLayout type.

Public Constructors

	Name	Description
	GridLayout	Initializes a new instance of the GridLayout class.

Public Properties

	Name	Description
	anchoring	Gets or sets how to align links to the anchor points of nodes. (Inherited from Layout.)
	endNode	Gets or sets the node that is placed at the lower right corner of the grid.
	gridSize	Gets or sets the desired distance between adjacent grid points.
	iterations	Gets or sets for how many iterations to run the grid layout algorithm.
	keepGroupLayout	Gets or sets whether to treat each group of attached nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object. (Inherited from Layout.)
	multipleGraphsPlacement	Gets or sets how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	startNode	Gets or sets the node that is placed at the upper left corner of the grid.
	timeLimit	Gets or sets a time-out value for the path-finding part of the grid layout algorithm.
	useLongestPath	Gets or sets a value indicating whether the algorithm should arrange nodes around the longest path in the graph.

Public Methods

	Name	Description
	arrange	Applies the layout to the specified graph.

See Also

GridLayout Class
MindFusion.Graphs Namespace

GridLayout Constructor

Initializes a new instance of the GridLayout class.

Namespace: MindFusion.Graphs
Assembly: GridLayout.js

Syntax

JavaScript Copy Code

```
function GridLayout ()
```

See Also

GridLayout Class
MindFusion.Graphs Namespace

GridLayout Methods

GridLayout.arrange Method

Applies the layout to the specified graph.

Namespace: MindFusion.Graphs
Assembly: GridLayout.js

Syntax

JavaScript — Copy Code

```
function arrange (graph)
```

Parameters

graph
Graph. The graph to arrange.

Remarks

This method is called internally by Diagram.arrange.

See Also

GridLayout Members
GridLayout Class
MindFusion.Graphs Namespace

GridLayout Properties

GridLayout.endNode Property

Gets or sets the node that is placed at the lower right corner of the grid.

Namespace: MindFusion.Graphs
Assembly: GridLayout.js

Syntax

JavaScript — Copy Code

```
get endNode() {}
set endNode(value) {}
```

Property Value

DiagramNode

An instance of the DiagramNode class. Default value is **null**.

Remarks

If you assign a node to this property, then startNode cannot be **null** and must be set too. GridLayout usually produces better layouts if both **startNode** and **endNode** are **null** references; in such a case the algorithm automatically selects suitable nodes.

See Also

GridLayout Members
GridLayout Class
MindFusion.Graphs Namespace

GridLayout.gridSize Property

Gets or sets the desired distance between adjacent grid points.

Namespace: MindFusion.Graphs
Assembly: GridLayout.js

Syntax

JavaScript Copy Code

```
get gridSize() {}
set gridSize(value) {}
```

Property Value

Number

A number specifying the grid size. The default is 25.

Remarks

When placing a node at a grid position, the node center point is aligned to the grid point at that position.

See Also

GridLayout Members
GridLayout Class
MindFusion.Graphs Namespace

GridLayout.iterations Property

Gets or sets for how many iterations to run the grid layout algorithm.

Namespace: MindFusion.Graphs
Assembly: GridLayout.js

Syntax

JavaScript Copy Code

```
get iterations() {}
set iterations(value) {}
```

Property Value

Number

An integer number specifying the number of iterations. The default is 3000.

Remarks

More iterations would yield better layouts but require more time to complete.

See Also

GridLayout Members
GridLayout Class
MindFusion.Graphs Namespace

GridLayout.startNode Property

Gets or sets the node that is placed at the upper left corner of the grid.

Namespace: MindFusion.Graphs
Assembly: GridLayout.js

Syntax

JavaScript Copy Code

```
get startNode() {}
set startNode(value) {}
```

Property Value

DiagramNode

An instance of the DiagramNode class. The default is **null**.

Remarks

If you assign a node to this property, then endNode cannot be **null** and must be set too. GridLayout usually produces better layouts if both **startNode** and **endNode** are **null** references; in such a case the algorithm automatically selects suitable nodes.

See Also

GridLayout Members
GridLayout Class
MindFusion.Graphs Namespace

GridLayout.timeLimit Property

Gets or sets a time-out value for the path-finding part of the grid layout algorithm.

Namespace: MindFusion.Graphs
Assembly: GridLayout.js

Syntax

JavaScript Copy Code

```
get timeLimit() {}
set timeLimit(value) {}
```

Property Value

Number

An integer value specifying time-out in milliseconds. The default is 10000 (10 seconds).

Remarks

The grid layout algorithm arranges graph nodes and links along the longest path in the graph. Large graphs with many links could contain hundreds of thousands of possible paths, so finding the longest one would be very time-consuming. When the time-out set with this property occurs, the path-finding routine stops and the layout algorithm uses the longest path known at that moment. Usually, for graphs containing several hundreds of nodes, 1-2 seconds are enough to find a path long enough to produce a nice layout.

See Also

GridLayout Members
GridLayout Class
MindFusion.Graphs Namespace
useLongestPath Property

GridLayout.useLongestPath Property

Gets or sets a value indicating whether the algorithm should arrange nodes around the longest path in the graph.

Namespace: MindFusion.Graphs
Assembly: GridLayout.js

Syntax

JavaScript Copy Code

```
get useLongestPath() {}
set useLongestPath(value) {}
```

Property Value

Boolean

true to look for the longest path in the graph and arrange nodes around it; otherwise, **false**.

Remarks

Initially, the algorithm places nodes and links from the longest path in the graph at the grid diagonal, and arranges the rest of the nodes around the diagonal. Large graphs with many links could contain hundreds of thousands of possible paths, so finding the longest one would be very time-consuming. Disabling this property will let you skip the path-finding part of the algorithm, which will make it work much faster.

See Also

GridLayout Members
GridLayout Class
MindFusion.Graphs Namespace
timeLimit Property

LayeredLayout Class

Implements a layered graph layout algorithm.

Syntax

JavaScript Copy Code

```
// class
LayeredLayout.prototype = {}
```

Remarks

This algorithm assigns diagram nodes to distinct horizontal or vertical layers. While arranging the layers, the layout routine seeks to meet the following criteria:

- connected nodes must be placed close together;
- links must flow in one direction if possible;
- links must cross as few layers as possible;
- links must not cross other links;

To apply this layout to a diagram, create a **LayeredLayout** instance, set its members and invoke the arrange method of the Diagram class. The method lays out all subgraphs in a diagram.

The final appearance of arranged graphs depends mostly on the direction value, which defines the overall orientation of the layout. The layerDistance and nodeDistance values specify how much space to leave between adjacent layers and adjacent nodes in a layer.

Example

The following example applies a **LayeredLayout** to a diagram:

JavaScript Copy Code

```
var layout = new MindFusion.Graphs.LayeredLayout();
layout.direction = MindFusion.Graphs.LayoutDirection.TopToBottom;
layout.siftingRounds = 3;
diagram.arrange(layout);
```

Inheritance Hierarchy

MindFusion.Graphs.Layout
  MindFusion.Graphs.LayeredLayout

See Also

LayeredLayout Members
MindFusion.Graphs Namespace

LayeredLayout Members

Implements a layered graph layout algorithm.

The following tables list the members exposed by the LayeredLayout type.

Public Constructors

	Name	Description
	LayeredLayout	Initializes a new instance of the **LayeredLayout** class.

Public Properties

	Name	Description
	anchoring	Gets or sets a value indicating how to align links to the anchor points of nodes.
	direction	Gets or sets the orientation of the arranged graph.
	endNode	Gets or sets the node that should be placed in the last layer of the arranged graph.
	keepGroupLayout	Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.
	layerDistance	Gets or sets the desired distance between layer axis lines.
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object. (Inherited from Layout.)
	multipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.
	nodeDistance	Gets or sets the space to leave between adjacent nodes on the same level.
	siftingRounds	Gets or sets how many times to perform pair-swapping iterations on each layer.
	startNode	Gets or sets the node that should be placed in the first layer of the arranged graph.

Public Methods

	Name	Description
	arrange	Applies the layout to the specified graph.

See Also

LayeredLayout Class
MindFusion.Graphs Namespace

LayeredLayout Constructor

Initializes a new instance of the LayeredLayout class.

Namespace: MindFusion.Graphs
Assembly: LayeredLayout.js

Syntax

JavaScript Copy Code

```
function LayeredLayout ()
```

See Also

LayeredLayout Class
MindFusion.Graphs Namespace

LayeredLayout Methods

LayeredLayout.arrange Method

Applies the layout to the specified graph.

Namespace: MindFusion.Graphs
Assembly: LayeredLayout.js

Syntax

JavaScript Copy Code

```
function arrange (graph)
```

Parameters

graph
The Graph to arrange.

Remarks

This method is called internally by Diagram.arrange and ContainerNode.arrange.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Graphs Namespace

LayeredLayout Properties

LayeredLayout.anchoring Property

Gets or sets a value indicating how to align links to the anchor points of nodes.

Namespace: MindFusion.Graphs
Assembly: LayeredLayout.js

Syntax

JavaScript Copy Code

```
get anchoring() {}
set anchoring(value) {}
```

Property Value

Anchoring

A member of the Anchoring enumeration. The default value is **Ignore**.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Graphs Namespace

LayeredLayout.direction Property

Gets or sets the orientation of the arranged graph.

Namespace: MindFusion.Graphs
Assembly: LayeredLayout.js

Syntax

JavaScript — Copy Code

```
get direction() {}
set direction(value) {}
```

Property Value

LayoutDirection

A member of the LayoutDirection enumeration. The default value is **TopToBottom**.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Graphs Namespace

LayeredLayout.endNode Property

Gets or sets the node that should be placed in the last layer of the arranged graph.

Namespace: MindFusion.Graphs
Assembly: LayeredLayout.js

Syntax

JavaScript — Copy Code

```
get endNode() {}
set endNode(value) {}
```

Property Value

DiagramNode

A DiagramNode representing the end node.

Remarks

This value is honored only if startNode is specified too.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Graphs Namespace

LayeredLayout.keepGroupLayout Property

Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.

Namespace: MindFusion.Graphs
Assembly: LayeredLayout.js

Syntax

JavaScript — Copy Code

```
get keepGroupLayout() {}
set keepGroupLayout(value) {}
```

Property Value

Boolean

true if each group of attached nodes should be treated as a single vertex in the arranged graph, or **false** otherwise.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Graphs Namespace

LayeredLayout.layerDistance Property

Gets or sets the desired distance between layer axis lines.

Namespace: MindFusion.Graphs
Assembly: LayeredLayout.js

Syntax

JavaScript Copy Code

```
get layerDistance() {}
set layerDistance(value) {}
```

Property Value

Number

The desired distance between layer axis lines.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Graphs Namespace

LayeredLayout.multipleGraphsPlacement Property

Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.

Namespace: MindFusion.Graphs
Assembly: LayeredLayout.js

Syntax

JavaScript Copy Code

```
get multipleGraphsPlacement() {}
set multipleGraphsPlacement(value) {}
```

Property Value

MultipleGraphsPlacement

A member of the MultipleGraphsPlacement enumeration. The default value is **Horizontal**.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Graphs Namespace

LayeredLayout.nodeDistance Property

Gets or sets the space to leave between adjacent nodes on the same level.

Namespace: MindFusion.Graphs
Assembly: LayeredLayout.js

Syntax

JavaScript Copy Code

```
get nodeDistance() {}
set nodeDistance(value) {}
```

Property Value

Number

The space to leave between adjacent nodes on the same level.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Graphs Namespace

LayeredLayout.siftingRounds Property

Gets or sets how many times to perform pair-swapping iterations on each layer.

Namespace: MindFusion.Graphs
Assembly: LayeredLayout.js

Syntax

JavaScript Copy Code

```
get siftingRounds() {}
set siftingRounds(value) {}
```

Property Value

Number

An integer value specifying how many pair-swapping iterations to perform. The default is 1.

Remarks

The algorithm swaps node pairs on each layer aiming to decrease the number of crossing links. Assigning a higher value to this property helps decrease the number of crossings, at the cost of longer execution times.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Graphs Namespace

LayeredLayout.startNode Property

Gets or sets the node that should be placed in the first layer of the arranged graph.

Namespace: MindFusion.Graphs
Assembly: LayeredLayout.js

Syntax

JavaScript Copy Code

```
get startNode() {}
set startNode(value) {}
```

Property Value

DiagramNode

A DiagramNode representing the start node.

Remarks

If there is no start node specified, the algorithm selects one automatically based on graph's topological order.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Graphs Namespace
endNode Property

Layout Class

Superclass of graph layout algorithms that defines their common properties.

Syntax

JavaScript Copy Code

```
// class
Layout.prototype = {}
```

Inheritance Hierarchy

MindFusion.Graphs.Layout
MindFusion.Graphs.BorderedTreeLayout
MindFusion.Graphs.FlowchartLayout
MindFusion.Graphs.FractalLayout
MindFusion.Graphs.GridLayout
MindFusion.Graphs.LayeredLayout
MindFusion.Graphs.OrthogonalLayout
MindFusion.Graphs.RadialTreeLayout
MindFusion.Graphs.SpringLayout
MindFusion.Graphs.TopologicalLayout
MindFusion.Graphs.TreeLayout
MindFusion.Graphs.TreeMapLayout

See Also

Layout Members
MindFusion.Graphs Namespace

Layout Members

Superclass of graph layout algorithms that defines their common properties.

The following tables list the members exposed by the Layout type.

Public Properties

	Name	Description
	anchoring	Gets or sets how to align links to the anchor points of nodes.
	keepGroupLayout	Gets or sets whether to treat each group of attached nodes as a single vertex in the arranged graph.
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object.
	multipleGraphsPlacement	Gets or sets how multiple independent graphs in the diagram should be positioned relatively to each other.

See Also

Layout Properties

Layout.anchoring Property

Gets or sets how to align links to the anchor points of nodes.

Namespace: MindFusion.Graphs
Assembly: Layout.js

Syntax

JavaScript Copy Code

```
get anchoring() {}
set anchoring(value) {}
```

Property Value

Anchoring

A member of the LayoutDirection enumeration.

See Also

Layout Members
Layout Class
MindFusion.Graphs Namespace

Layout.keepGroupLayout Property

Gets or sets whether to treat each group of attached nodes as a single vertex in the arranged graph.

Namespace: MindFusion.Graphs
Assembly: Layout.js

Syntax

JavaScript Copy Code

```
get keepGroupLayout() {}
set keepGroupLayout(value) {}
```

Property Value

Boolean

true if each group should be processed as an integral graph node, or false otherwise. The default is false.

See Also

Layout Members
Layout Class
MindFusion.Graphs Namespace

Layout.margins Property

Gets or sets the margins' size around individual subgraphs processed by this layout object.

Namespace: MindFusion.Graphs
Assembly: Layout.js

Syntax

JavaScript Copy Code

```
get margins() {}
set margins(value) {}
```

Property Value

Number

A number specifying size of margins around arranged graphs.

See Also

Layout Members
Layout Class
MindFusion.Graphs Namespace

Layout.multipleGraphsPlacement Property

Gets or sets how multiple independent graphs in the diagram should be positioned relatively to each other.

Namespace: MindFusion.Graphs
Assembly: Layout.js

Syntax

JavaScript Copy Code

```
get multipleGraphsPlacement() {}
set multipleGraphsPlacement(value) {}
```

Property Value

MultipleGraphsPlacement

A member of the MultipleGraphsPlacement enumeration.

See Also

Layout Members
Layout Class
MindFusion.Graphs Namespace

LayoutDirection Enumeration

Specifies in what direction to place nodes processed by a layout algorithm.

Namespace: MindFusion.Graphs
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
LayoutDirection = {}
```

Members

	Member name	Description
	TopToBottom	Indicates a top-to-bottom layout direction.
	LeftToRight	Indicates a left-to-right layout direction.
	BottomToTop	Indicates a bottom-to-top layout direction.
	RightToLeft	Indicates a right-to-left layout direction.

Remarks

Members of this enumeration can be assigned to the direction property of layout classes.

See Also

MindFusion.Graphs Namespace

MultipleGraphsPlacement Enumeration

Specifies placement of graph connected components relatively to each other.

Namespace: MindFusion.Graphs
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
MultipleGraphsPlacement = {}
```

Members

	Member name	Description
	Vertical	Indicates that subgraphs should be placed in a column.
	Horizontal	Indicates that subgraphs should be placed in a row.
	MinimalArea	Indicates that subgraphs should be packed in a square-like area.

Remarks

Members of this enumeration can be assigned to the multipleGraphsPlacement property of layout classes.

See Also

MindFusion.Graphs Namespace

Orientation Enumeration

Specifies general layout orientation.

Namespace: MindFusion.Graphs
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
Orientation = {}
```

Members

	Member name	Description
	Vertical	Specifies vertical orientation.
	Horizontal	Specifies horizonal orientation.

Remarks

Members of this enumeration can be assigned to the orientation property of layout classes.

See Also

MindFusion.Graphs Namespace

OrthogonalLayout Class

Implements orthogonal graph layout algorithm.Each link is drawn as a chain of alternating horizontal and vertical segments.Nodes are placed in a way that facilitates few links bends and crossings.

Syntax

JavaScript Copy Code

```
// class
OrthogonalLayout.prototype = {}
```

Inheritance Hierarchy

MindFusion.Graphs.Layout
  MindFusion.Graphs.OrthogonalLayout

See Also

OrthogonalLayout Members
MindFusion.Graphs Namespace

OrthogonalLayout Members

Implements orthogonal graph layout algorithm.Each link is drawn as a chain of alternating horizontal and vertical segments.Nodes are placed in a way that facilitates few links bends and crossings.

The following tables list the members exposed by the OrthogonalLayout type.

Public Constructors

	Name	Description
	OrthogonalLayout	Initializes a new instance of the OrthogonalLayout class.

Public Properties

	Name	Description
	anchoring	Gets or sets how to align links to the anchor points of nodes. (Inherited from Layout.)
	keepGroupLayout	Gets or sets whether to treat each group of attached nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object. (Inherited from Layout.)
	minLaneSize	Gets or sets a value indicating the minimum size of a lane.
	multipleGraphsPlacement	Gets or sets how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	padding	Gets or sets a value indicating how much space to leave between nodes in adjacent lanes.
	refine	Gets or sets a value indicating whether to refine the layout.

Public Methods

	Name	Description

	arrange	Applies the layout to the specified graph.

See Also

OrthogonalLayout Class
MindFusion.Graphs Namespace

OrthogonalLayout Constructor

Initializes a new instance of the OrthogonalLayout class.

Namespace: MindFusion.Graphs
Assembly: OrthogonalLayout.js

Syntax

JavaScript Copy Code

```
function OrthogonalLayout ()
```

See Also

OrthogonalLayout Class
MindFusion.Graphs Namespace

OrthogonalLayout Methods

OrthogonalLayout.arrange Method

Applies the layout to the specified graph.

Namespace: MindFusion.Graphs
Assembly: OrthogonalLayout.js

Syntax

JavaScript Copy Code

```
function arrange (graph)
```

Parameters

graph
Graph. The graph to arrange.

See Also

OrthogonalLayout Members
OrthogonalLayout Class
MindFusion.Graphs Namespace

OrthogonalLayout Properties

OrthogonalLayout.minLaneSize Property

Gets or sets a value indicating the minimum size of a lane.

Namespace: MindFusion.Graphs
Assembly: OrthogonalLayout.js

Syntax

JavaScript Copy Code

```
get minLaneSize() {}
set minLaneSize(value) {}
```

Property Value

Number

The minimum size of a lane.

See Also

OrthogonalLayout Members
OrthogonalLayout Class
MindFusion.Graphs Namespace

OrthogonalLayout.padding Property

Gets or sets a value indicating how much space to leave between nodes in adjacent lanes.

Namespace: MindFusion.Graphs
Assembly: OrthogonalLayout.js

Syntax

JavaScript Copy Code

```
get padding() {}
set padding(value) {}
```

Property Value

Number

The space to leave between nodes in adjacent lanes.

See Also

OrthogonalLayout Members
OrthogonalLayout Class
MindFusion.Graphs Namespace

OrthogonalLayout.refine Property

Gets or sets a value indicating whether to refine the layout.

Namespace: MindFusion.Graphs
Assembly: OrthogonalLayout.js

Syntax

JavaScript Copy Code

```
get refine() {}
set refine(value) {}
```

Property Value

Boolean

True to refine the layout, or false otherwise.

See Also

OrthogonalLayout Members
OrthogonalLayout Class
MindFusion.Graphs Namespace

Path Class

Initializes a new instance of the Path class.

Syntax

JavaScript Copy Code

```
// class
Path.prototype = {}
```

Inheritance Hierarchy

MindFusion.Graphs.Path

See Also

Path Members
MindFusion.Graphs Namespace

Path Members

Initializes a new instance of the Path class.

The following tables list the members exposed by the Path type.

Public Constructors

	Name	Description
	Path	Initializes a new instance of the Path class.

See Also

Path Class
MindFusion.Graphs Namespace

Path Constructor

Initializes a new instance of the Path class.

Namespace: MindFusion.Graphs
Assembly: Path.js

Syntax

JavaScript Copy Code

```
function Path ()
```

See Also

Path Class
MindFusion.Graphs Namespace

PathFinder Class

Initializes a new instance of the PathFinder class.

Syntax

JavaScript Copy Code

```
// class
PathFinder.prototype = {}
```

Inheritance Hierarchy

MindFusion.Graphs.PathFinder

See Also

PathFinder Members
MindFusion.Graphs Namespace

PathFinder Members

Initializes a new instance of the PathFinder class.

The following tables list the members exposed by the PathFinder type.

Public Constructors

	Name	Description
	PathFinder	Initializes a new instance of the PathFinder class.

Public Methods

	Name	Description
	findAllPaths	Finds and returns all paths starting from node 'from' andending at node 'to'. Returns empty collection if nopath exists.

See Also

PathFinder Class
MindFusion.Graphs Namespace

PathFinder Constructor

Initializes a new instance of the PathFinder class.

Namespace: MindFusion.Graphs
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
function PathFinder ()
```

See Also

PathFinder Class
MindFusion.Graphs Namespace

PathFinder Methods

PathFinder.findAllPaths Method

Finds and returns all paths starting from node 'from' andending at node 'to'. Returns empty collection if nopath exists.

Namespace: MindFusion.Graphs
Assembly: PathFinder.js

Syntax

JavaScript Copy Code

```
function findAllPaths ()
```

See Also

PathFinder Members
PathFinder Class
MindFusion.Graphs Namespace

RadialTreeLayout Class

Implements a radial tree layout algorithm.

Syntax

JavaScript Copy Code

```
// class
RadialTreeLayout.prototype = {}
```

Remarks

This layout class arranges tree levels in concentric circles around the root. The levelDistance property specifies the distance between adjacent circles. The direction property specifies placement of first child node. Set stretchFactor to a value larger than 1 in order to stretch level circles into ellipses, providing more space for the layout when using wider nodes.

Inheritance Hierarchy

MindFusion.Graphs.Layout
  MindFusion.Graphs.RadialTreeLayout

See Also

RadialTreeLayout Members
MindFusion.Graphs Namespace

RadialTreeLayout Members

Implements a radial tree layout algorithm.

The following tables list the members exposed by the RadialTreeLayout type.

Public Constructors

	Name	Description
	RadialTreeLayout	Initializes a new instance of the RadialTreeLayout class.

Public Properties

	Name	Description
	anchoring	Gets or sets how to align links to the anchor points of nodes. (Inherited from Layout.)
	direction	Gets or sets the orientation of the arranged graph.
	keepGroupLayout	Gets or sets whether to treat each group of attached nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	keepRootPosition	Gets or sets a value indicating whether to keep the root node at its original position.
	levelDistance	Gets or sets the space to leave between adjacent levels of the tree.
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object. (Inherited from Layout.)
	multipleGraphsPlacement	Gets or sets how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	reversedLinks	Gets or sets a value indicating the link direction that relates parent nodes to child nodes.
	stretchFactor	Gets or sets the stretch factor in radial layouts.

Public Methods

	Name	Description
	arrange	Applies the layout to the specified graph.

See Also

RadialTreeLayout Class
MindFusion.Graphs Namespace

RadialTreeLayout Constructor

Initializes a new instance of the RadialTreeLayout class.

Namespace: MindFusion.Graphs
Assembly: RadialTreeLayout.js

Syntax

JavaScript Copy Code

```
function RadialTreeLayout ()
```

See Also

RadialTreeLayout Class
MindFusion.Graphs Namespace

RadialTreeLayout Methods

RadialTreeLayout.arrange Method

Applies the layout to the specified graph.

Namespace: MindFusion.Graphs
Assembly: RadialTreeLayout.js

Syntax

JavaScript Copy Code

```
function arrange (graph)
```

Parameters

graph
Graph. The graph to arrange.

See Also

RadialTreeLayout Members
RadialTreeLayout Class
MindFusion.Graphs Namespace

RadialTreeLayout Properties

RadialTreeLayout.direction Property

Gets or sets the orientation of the arranged graph.

Namespace: MindFusion.Graphs
Assembly: RadialTreeLayout.js

Syntax

JavaScript Copy Code

```
get direction() {}
set direction(value) {}
```

Property Value

LayoutDirection

A member of the LayoutDirection enumeration.

Remarks

In radial layouts, the **direction** property specifies placement of first child node relatively to the root.

See Also

RadialTreeLayout Members
RadialTreeLayout Class
MindFusion.Graphs Namespace

RadialTreeLayout.keepRootPosition Property

Gets or sets a value indicating whether to keep the root node at its original position.

Namespace: MindFusion.Graphs
Assembly: RadialTreeLayout.js

Syntax

JavaScript Copy Code

```
get keepRootPosition() {}
set keepRootPosition(value) {}
```

Property Value

Boolean

true if the root node should be kept at its original position, or **false** otherwise.

See Also

RadialTreeLayout Members
RadialTreeLayout Class
MindFusion.Graphs Namespace

RadialTreeLayout.levelDistance Property

Gets or sets the space to leave between adjacent levels of the tree.

Namespace: MindFusion.Graphs
Assembly: RadialTreeLayout.js

Syntax

JavaScript Copy Code

```
get levelDistance() {}
set levelDistance(value) {}
```

Property Value

Number

The space to leave between adjacent levels of the tree.

See Also

RadialTreeLayout Members
RadialTreeLayout Class
MindFusion.Graphs Namespace

RadialTreeLayout.reversedLinks Property

Gets or sets a value indicating the link direction that relates parent nodes to child nodes.

Namespace: MindFusion.Graphs
Assembly: RadialTreeLayout.js

Syntax

JavaScript Copy Code

```
get reversedLinks() {}
set reversedLinks(value) {}
```

Property Value

Boolean

true to indicate reversed direction (links point to parent nodes), or **false** otherwise.

See Also

RadialTreeLayout Members
RadialTreeLayout Class
MindFusion.Graphs Namespace

RadialTreeLayout.stretchFactor Property

Gets or sets the stretch factor in radial layouts.

Namespace: MindFusion.Graphs
Assembly: RadialTreeLayout.js

Syntax

JavaScript Copy Code

```
get stretchFactor() {}
set stretchFactor(value) {}
```

Property Value

Number

A stretch factor that lets you set ellipse eccentricity. The default value is 1.

Remarks

Set **stretchFactor** to a value larger than 1 in order to stretch level circles into ellipses, providing more space for the layout when using wider nodes.

See Also

RadialTreeLayout Members
RadialTreeLayout Class
MindFusion.Graphs Namespace

SpringLayout Class

Implements the Spring-Embedder graph layout algorithm.

Syntax

JavaScript Copy Code

```
// class
SpringLayout.prototype = {}
```

Remarks

To apply this layout to a diagram, create a **SpringLayout** instance, set its members and invoke the arrange method of the Diagram class. The method lays out all subgraphs in the diagram.

Spring-Embedder produces layouts having uniform distribution of nodes by simulating a physical system in which nodes repulse each other and the links between them act as confining springs. Nodes are moved around in an iterative process. The forces that act on a node are calculated, taking in account the positions of surrounding nodes and links from the previous iteration.

The final appearance of arranged graphs depends on the nodeDistance value that specifies how nodes are distributed by the layout routine. The layout routine runs as much iterations as set in iterations, calculating better node positions at each step.

Example

The following example applies a **SpringLayout** to a diagram:

JavaScript Copy Code

```
var layout = new MindFusion.Graphs.SpringLayout();
layout.iterations = 100;
diagram.arrange(layout);
```

Inheritance Hierarchy

MindFusion.Graphs.Layout
  MindFusion.Graphs.SpringLayout

See Also

SpringLayout Members
MindFusion.Graphs Namespace

SpringLayout Members

Implements the Spring-Embedder graph layout algorithm.

The following tables list the members exposed by the SpringLayout type.

Public Constructors

	Name	Description
	SpringLayout	Initializes a new instance of the **SpringLayout** class.

Public Properties

	Name	Description
	anchoring	Gets or sets a value indicating how to align links to the anchor points of nodes.
	iterations	Gets or sets the number of iterations to run the layout routine.
	keepGroupLayout	Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object. (Inherited from Layout.)
	multipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.
	nodeDistance	Gets or sets the desired distance between nodes.

Public Methods

	Name	Description
	arrange	Applies the layout to the specified graph.

See Also

SpringLayout Class
MindFusion.Graphs Namespace

SpringLayout Constructor

Initializes a new instance of the SpringLayout class.

Namespace: MindFusion.Graphs
Assembly: SpringLayout.js

Syntax

JavaScript Copy Code

```
function SpringLayout ()
```

See Also

SpringLayout Class
MindFusion.Graphs Namespace

SpringLayout Methods

SpringLayout.arrange Method

Applies the layout to the specified graph.

Namespace: MindFusion.Graphs
Assembly: SpringLayout.js

Syntax

JavaScript — Copy Code

```
function arrange (graph)
```

Parameters

graph
The Graph to arrange.

Remarks

This method is called internally by Diagram.arrange and ContainerNode.arrange.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Graphs Namespace

SpringLayout Properties

SpringLayout.anchoring Property

Gets or sets a value indicating how to align links to the anchor points of nodes.

Namespace: MindFusion.Graphs
Assembly: SpringLayout.js

Syntax

JavaScript — Copy Code

```
get anchoring() {}
set anchoring(value) {}
```

Property Value

Anchoring

A member of the Anchoring enumeration. The default value is **Ignore**.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Graphs Namespace

SpringLayout.iterations Property

Gets or sets the number of iterations to run the layout routine.

Namespace: MindFusion.Graphs
Assembly: SpringLayout.js

Syntax

JavaScript — Copy Code

```
get iterations() {}
set iterations(value) {}
```

Property Value

Number

The number of iterations to run the layout routine.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Graphs Namespace

SpringLayout.keepGroupLayout Property

Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.

Namespace: MindFusion.Graphs
Assembly: SpringLayout.js

Syntax

JavaScript Copy Code

```
get keepGroupLayout() {}
set keepGroupLayout(value) {}
```

Property Value

Boolean

true if each group of attached nodes should be treated as a single vertex in the arranged graph, or **false** otherwise.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Graphs Namespace

SpringLayout.multipleGraphsPlacement Property

Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.

Namespace: MindFusion.Graphs
Assembly: SpringLayout.js

Syntax

JavaScript Copy Code

```
get multipleGraphsPlacement() {}
set multipleGraphsPlacement(value) {}
```

Property Value

MultipleGraphsPlacement

A member of the MultipleGraphsPlacement enumeration. The default value is **Horizontal**.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Graphs Namespace

SpringLayout.nodeDistance Property

Gets or sets the desired distance between nodes.

Namespace: MindFusion.Graphs
Assembly: SpringLayout.js

Syntax

JavaScript — Copy Code

```
get nodeDistance() {}
set nodeDistance(value) {}
```

Property Value

Number

The desired distance between nodes.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Graphs Namespace

TopologicalLayout Class

Applies topological ordering to the graph. This layout makes it easy to discern graph features such as overall flow direction, cycles and nested cycles.

Syntax

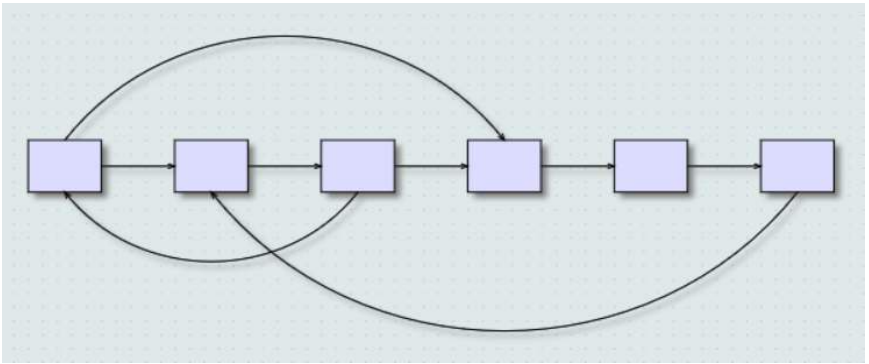
JavaScript — Copy Code

```
// class
TopologicalLayout.prototype = {}
```

Remarks

Topological ordering of graph nodes is one where link's origin node is placed before link's destination. TopologicalLayout arranges nodes in a row or a column (depending on the value of the direction property) in such a way that there are no backward links when the graph is acyclic. If the graph contains cycles, the algorithms selects ordering with as few backward links as possible.

Links that connect non-adjacent nodes are rendered as arcs. Links connecting adjacent nodes can be rendered either as arcs or as straight lines, depending on the bendAdjacentLinks property. The amplitude of link arcs is proportional to the distance between respective connected nodes. All forward links are rendered at one side of the nodes and all back links are rendered at the opposite side. Thus, all bent links follow the same rotation direction.



Example

The following example applies TopologicalLayout to the diagram:

JavaScript — Copy Code

```
var layout = new MindFusion.Graphs.TopologicalLayout();
layout.direction = MindFusion.Graphs.LayoutDirection.LeftToRight;
```

```
diagram.arrange(layout);
diagram.resizeToFitItems();
```

Inheritance Hierarchy

MindFusion.Graphs.Layout
  MindFusion.Graphs.TopologicalLayout

See Also

TopologicalLayout Members
MindFusion.Graphs Namespace

TopologicalLayout Members

Applies topological ordering to the graph. This layout makes it easy to discern graph features such as overall flow direction, cycles and nested cycles.

The following tables list the members exposed by the TopologicalLayout type.

Public Constructors

	Name	Description
	TopologicalLayout	Initializes a new instance of the **TopologicalLayout** class.

Public Properties

	Name	Description
	anchoring	Gets or sets how to align links to the anchor points of nodes. (Inherited from Layout.)
	bendAdjacentLinks	Gets or sets a value indicating whether to bend links connecting adjacent nodes or draw them as straight lines.
	direction	Gets or sets the orientation of the arranged graph.
	keepGroupLayout	Gets or sets whether to treat each group of attached nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object. (Inherited from Layout.)
	multipleGraphsPlacement	Gets or sets how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	nodeDistance	Gets or sets the desired distance between adjacent nodes.

Public Methods

	Name	Description
	arrange	Applies the layout to the specified graph.
	init	Returns a Builder object used to configure this **TopologicalLayout** instance.
	With	Returns a Builder object used to configure and create new **TopologicalLayout** instances.

See Also

TopologicalLayout Class
MindFusion.Graphs Namespace

TopologicalLayout Constructor

Initializes a new instance of the TopologicalLayout class.

Namespace: MindFusion.Graphs
Assembly: TopologicalLayout.js

Syntax

JavaScript — Copy Code

```
function TopologicalLayout ()
```

See Also

TopologicalLayout Class
MindFusion.Graphs Namespace

TopologicalLayout Methods

TopologicalLayout.arrange Method

Applies the layout to the specified graph.

Namespace: MindFusion.Graphs
Assembly: TopologicalLayout.js

Syntax

JavaScript — Copy Code

```
function arrange (graph)
```

Parameters

graph
The Graph to arrange.

Remarks

This method is called internally by Diagram.arrange and ContainerNode.arrange.

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Graphs Namespace

TopologicalLayout.init Method

Returns a Builder object used to configure this TopologicalLayout instance.

Namespace: MindFusion.Graphs
Assembly: TopologicalLayout.js

Syntax

JavaScript — Copy Code

```
function init ()
```

Return Value

A Builder instance.

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Graphs Namespace

TopologicalLayout.With Method

Returns a Builder object used to configure and create new TopologicalLayout instances.

Namespace: MindFusion.Graphs
Assembly: TopologicalLayout.js

Syntax

JavaScript Copy Code

```
function With ()
```

Return Value

A Builder instance.

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Graphs Namespace

TopologicalLayout Properties

TopologicalLayout.bendAdjacentLinks Property

Gets or sets a value indicating whether to bend links connecting adjacent nodes or draw them as straight lines.

Namespace: MindFusion.Graphs
Assembly: TopologicalLayout.js

Syntax

JavaScript Copy Code

```
get bendAdjacentLinks() {}
set bendAdjacentLinks(value) {}
```

Property Value

Boolean

true to bend links connecting adjacent nodes, or false otherwise.

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Graphs Namespace

TopologicalLayout.direction Property

Gets or sets the orientation of the arranged graph.

Namespace: MindFusion.Graphs
Assembly: TopologicalLayout.js

Syntax

JavaScript Copy Code

```
get direction() {}
set direction(value) {}
```

Property Value

LayoutDirection

A member of the LayoutDirection enumeration.

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Graphs Namespace

TopologicalLayout.nodeDistance Property

Gets or sets the desired distance between adjacent nodes.

Namespace: MindFusion.Graphs
Assembly: TopologicalLayout.js

Syntax

JavaScript Copy Code

```
get nodeDistance() {}
set nodeDistance(value) {}
```

Property Value

Number

The desired distance between adjacent nodes.

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Graphs Namespace

TreeLayout Class

Implements algorithms for arranging tree structures.

Syntax

JavaScript Copy Code

```
// class
TreeLayout.prototype = {}
```

Remarks

To apply this layout to a diagram, create a **TreeLayout** instance, set its members and invoke the arrange method of the Diagram class. The method lays out all subtrees in a diagram.

As long as the graph is a tree, the algorithm can find the root node automatically. The distance between the root and the second level and between all subsequent levels can be set through the levelDistance field. The distance between adjacent nodes in the same level is set through nodeDistance.

In addition to arranging the tree nodes, **TreeLayout** also sets the shape of links as set through its linkType property.

Example

The following example applies a **TreeLayout** to a diagram:

JavaScript Copy Code

```
var layout = new MindFusion.Graphs.TreeLayout();
layout.direction = MindFusion.Graphs.LayoutDirection.TopToBottom;
linkType = MindFusion.Graphs.TreeLayoutLinkType.Cascading;
diagram.arrange(layout);
```

Inheritance Hierarchy

MindFusion.Graphs.Layout
  MindFusion.Graphs.TreeLayout

See Also

TreeLayout Members
MindFusion.Graphs Namespace

TreeLayout Members

Implements algorithms for arranging tree structures.

The following tables list the members exposed by the TreeLayout type.

Public Constructors

	Name	Description
	TreeLayout	Initializes a new instance of the **TreeLayout** class.

Public Properties

	Name	Description
	anchoring	Gets or sets a value indicating how to align links to the anchor points of nodes.
	compactAssistants	Gets or sets a value indicating whether the "assistant" nodes on the same side of a single parentare arranged as close to each other as possible.
	direction	Gets or sets the orientation of the arranged graph.
	enableAssistants	Gets or sets a value Indicating whether the "assistant" trait is regarded when performing the layout.
	keepGroupLayout	Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.
	keepRootPosition	Gets or sets a value indicating whether to keep the root node at its original position.
	levelDistance	Gets or sets the space to leave between adjacent levels of the tree.
	linkType	Gets or sets the type of the links in the arranged tree.
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object. (Inherited from Layout.)
	multipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.
	nodeDistance	Gets or sets the space to leave between adjacent nodes on the same level.
	reversedLinks	Gets or sets a value indicating the link direction that relates parent nodes to child nodes.

Public Methods

	Name	Description
	arrange	Applies the layout to the specified graph.

See Also

TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout Constructor

Initializes a new instance of the TreeLayout class.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
function TreeLayout ()
```

See Also

TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout Methods

TreeLayout.arrange Method

Applies the layout to the specified graph.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
function arrange (graph)
```

Parameters

graph
The Graph to arrange.

Remarks

This method is called internally by Diagram.arrange and ContainerNode.arrange.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout Properties

TreeLayout.anchoring Property

Gets or sets a value indicating how to align links to the anchor points of nodes.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
get anchoring() {}
set anchoring(value) {}
```

Property Value

Anchoring

A member of the Anchoring enumeration. The default value is **Ignore**.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout.compactAssistants Property

Gets or sets a value indicating whether the "assistant" nodes on the same side of a single parentare arranged as close to each other as possible.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
get compactAssistants() {}
set compactAssistants(value) {}
```

Property Value

Boolean

true if the "assistant" nodes on the same side of a single parentare arranged as close to each other as possible, or false otherwise.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout.direction Property

Gets or sets the orientation of the arranged graph.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
get direction() {}
set direction(value) {}
```

Property Value

LayoutDirection

A member of the LayoutDirection enumeration. The default value is **TopToBottom**.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout.enableAssistants Property

Gets or sets a value Indicating whether the "assistant" trait is regarded when performing the layout.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
get enableAssistants() {}
set enableAssistants(value) {}
```

Property Value

Boolean

true if the "assistant" trait is regarded when performing the layout, or false otherwise.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout.keepGroupLayout Property

Gets or sets a value indicating whether to treat each group of nodes as a single vertex in the arranged graph.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
get keepGroupLayout() {}
set keepGroupLayout(value) {}
```

Property Value

Boolean

true if each group of attached nodes should be treated as a single vertex in the arranged graph, or **false** otherwise.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout.keepRootPosition Property

Gets or sets a value indicating whether to keep the root node at its original position.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
get keepRootPosition() {}
set keepRootPosition(value) {}
```

Property Value

Boolean

true if the root node should be kept at its original position, or **false** otherwise.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout.levelDistance Property

Gets or sets the space to leave between adjacent levels of the tree.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
get levelDistance() {}
set levelDistance(value) {}
```

Property Value

Number

The space to leave between adjacent levels of the tree.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout.linkType Property

Gets or sets the type of the links in the arranged tree.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
get linkType() {}
set linkType(value) {}
```

Property Value

TreeLayoutLinkType

A member of the TreeLayoutLinkType enumeration. The default value is **Default**.

See Also

TreeLayout.multipleGraphsPlacement Property

Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
get multipleGraphsPlacement() {}
set multipleGraphsPlacement(value) {}
```

Property Value

MultipleGraphsPlacement

A member of the MultipleGraphsPlacement enumeration.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout.nodeDistance Property

Gets or sets the space to leave between adjacent nodes on the same level.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
get nodeDistance() {}
set nodeDistance(value) {}
```

Property Value

Number

The space to leave between adjacent nodes on the same level.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Graphs Namespace

TreeLayout.reversedLinks Property

Gets or sets a value indicating the link direction that relates parent nodes to child nodes.

Namespace: MindFusion.Graphs
Assembly: TreeLayout.js

Syntax

JavaScript Copy Code

```
get reversedLinks() {}
set reversedLinks(value) {}
```

Property Value

Boolean

true to indicate reversed direction (links point to parent nodes), or **false** otherwise.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Graphs Namespace

TreeLayoutLinkType Enumeration

Specifies the shape of the diagram links after they are laid out.

Namespace: MindFusion.Graphs
Assembly: Enum.js

Syntax

JavaScript Copy Code

```
// enum
TreeLayoutLinkType = {}
```

Members

	Member name	Description
	Default	Indicates that links will be arranged to point to the centers of the related nodes.
	Straight	Indicates that links will be attached to the middles of the adjoining node sides.
	Cascading	Indicates that links will be arranged as cascading and attached to the middles of the adjoining node sides.

Remarks

Members of this enumeration can be assigned to the linkType property of TreeLayout.

See Also

MindFusion.Graphs Namespace

TreeMapLayout Class

Performs tree-map layout on a graph.

Syntax

JavaScript Copy Code

```
// class
TreeMapLayout.prototype = {}
```

Remarks

Tree maps represent hierarchies by nesting child nodes within their parents, where the areas of leaf nodes are proportional to their Weight values. Unlike other layout algorithms, **TreeMapLayout** expects hierarchies to be defined via grouping or containment (see attachTo method and ContainerNode class), and will ignore any links in the diagram.

The diagram area covered by the topmost nodes in a hierarchy is specified via the layoutArea property. By default, the layout tries to keep the ratio of node sides as close as possible to one. However this could make it hard to distinguish separate levels of the hierarchy. To alleviate that, set squarify to **false**, and child nodes will be arranged either as a row or a column inside their parent node, alternating directions for each level. The drawback is that when weight ratios differ greatly or nodes contain many children, some nodes could end up with very narrow rectangles.

The padding size between parent nodes and their outermost children is specified via the containerPadding property. The padding size between child nodes is specified via padding.

Inheritance Hierarchy

MindFusion.Graphs.Layout
  MindFusion.Graphs.TreeMapLayout

See Also

TreeMapLayout Members
MindFusion.Graphs Namespace

TreeMapLayout Members

Performs tree-map layout on a graph.

The following tables list the members exposed by the TreeMapLayout type.

Public Constructors

	Name	Description
	TreeMapLayout	Initializes a new instance of the **TreeMapLayout** class.

Public Properties

	Name	Description
	anchoring	Gets or sets how to align links to the anchor points of nodes. (Inherited from Layout.)
	containerPadding	Gets or sets the padding inside containers.
	keepGroupLayout	Gets or sets whether to treat each group of attached nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	layoutArea	Gets or sets the rectangle in which the layout will try to arrange nodes.
	margins	Gets or sets the margins' size around individual subgraphs processed by this layout object. (Inherited from Layout.)
	multipleGraphsPlacement	Gets or sets how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	orientation	Gets or sets the orientation of the arranged graph.
	padding	Gets or sets the distance between adjacent nodes.
	squarify	Gets or sets a value indicating whether the layout should attempt to keep the dimension ratio of nodes closer to 1.

Public Methods

	Name	Description
	arrange	Applies the layout to the specified graph.

See Also

TreeMapLayout Class
MindFusion.Graphs Namespace

TreeMapLayout Constructor

Initializes a new instance of the TreeMapLayout class.

Namespace: MindFusion.Graphs
Assembly: TreeMapLayout.js

Syntax

JavaScript Copy Code

```
function TreeMapLayout ()
```

See Also

TreeMapLayout Class
MindFusion.Graphs Namespace

TreeMapLayout Methods

TreeMapLayout.arrange Method

Applies the layout to the specified graph.

Namespace: MindFusion.Graphs
Assembly: TreeMapLayout.js

Syntax

JavaScript Copy Code

```
function arrange (graph)
```

Parameters

graph
The Graph to arrange.

Remarks

This method is called internally by Diagram.arrange and ContainerNode.arrange.

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Graphs Namespace

TreeMapLayout Properties

TreeMapLayout.containerPadding Property

Gets or sets the padding inside containers.

Namespace: MindFusion.Graphs
Assembly: TreeMapLayout.js

Syntax

JavaScript Copy Code

```
get containerPadding() {}
set containerPadding(value) {}
```

Property Value

Number

The padding inside containers.

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Graphs Namespace

TreeMapLayout.layoutArea Property

Gets or sets the rectangle in which the layout will try to arrange nodes.

Namespace: MindFusion.Graphs
Assembly: TreeMapLayout.js

Syntax

JavaScript Copy Code

```
get layoutArea() {}
set layoutArea(value) {}
```

Property Value

Rect

A Rect instance specifying the area in which the layout will try to arrange nodes.

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Graphs Namespace

TreeMapLayout.orientation Property

Gets or sets the orientation of the arranged graph.

Namespace: MindFusion.Graphs
Assembly: TreeMapLayout.js

Syntax

JavaScript Copy Code

```
get orientation() {}
set orientation(value) {}
```

Property Value

Orientation

A member of the Orientation enumeration.

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Graphs Namespace

TreeMapLayout.padding Property

Gets or sets the distance between adjacent nodes.

Namespace: MindFusion.Graphs
Assembly: TreeMapLayout.js

Syntax

JavaScript Copy Code

```
get padding() {}
set padding(value) {}
```

Property Value

Number

The distance between adjacent nodes.

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Graphs Namespace

TreeMapLayout.squarify Property

Gets or sets a value indicating whether the layout should attempt to keep the dimension ratio of nodes closer to 1.

Namespace: MindFusion.Graphs
Assembly: TreeMapLayout.js

Syntax

JavaScript Copy Code

```
get squarify() {}
set squarify(value) {}
```

Property Value

Boolean

true if the layout should attempt to keep the dimension ratio of nodes closer to 1, or **false** otherwise.

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Graphs Namespace

Vertex Class

Represents a vertex in a graph.

Syntax

JavaScript Copy Code

```
// class
Vertex.prototype = {}
```

Inheritance Hierarchy

MindFusion.Graphs.Vertex

See Also

Vertex Members
MindFusion.Graphs Namespace

Vertex Members

Represents a vertex in a graph.

The following tables list the members exposed by the Vertex type.

Public Constructors

	Name	Description
	Vertex	Initializes a new instance of the **Vertex** class.

Public Fields

	Name	Description
	edges	A list containing all edges incident with this vertex.
	inEdges	A list containing all incoming edges.
	outEdges	A list containing all outgoing edges.
	weight	The relative weight of this vertex.

Public Methods

	Name	Description
	adjacentTo	Checks if there is an edge connecting this vertex with the specified vertex.
	degree	Returns the number of incident edges.
	getCommonEdge	Finds the edge connecting the current vertex with the specified vertex.
	getNeighbors	Returns an array containing all vertices adjacent to the current vertex.
	incidentWith	Checks if the current vertex is incident with the specified edge.

See Also

Vertex Class
MindFusion.Graphs Namespace

Vertex Fields

Vertex.edges Field

A list containing all edges incident with this vertex.

Namespace: MindFusion.Graphs
Assembly: Vertex.js

Syntax

JavaScript Copy Code

```
var edges
```

See Also

Vertex Members
Vertex Class
MindFusion.Graphs Namespace

Vertex.inEdges Field

A list containing all incoming edges.

Namespace: MindFusion.Graphs
Assembly: Vertex.js

Syntax

JavaScript Copy Code

```
var inEdges
```

See Also

Vertex Members
Vertex Class
MindFusion.Graphs Namespace

Vertex.outEdges Field

A list containing all outgoing edges.

Namespace: MindFusion.Graphs
Assembly: Vertex.js

Syntax

JavaScript Copy Code

```
var outEdges
```

See Also

Vertex Members
Vertex Class
MindFusion.Graphs Namespace

Vertex.weight Field

The relative weight of this vertex.

Namespace: MindFusion.Graphs
Assembly: Vertex.js

Syntax

JavaScript Copy Code

```
var weight
```

See Also

Vertex Members
Vertex Class
MindFusion.Graphs Namespace

Vertex Constructor

Initializes a new instance of the Vertex class.

Namespace: MindFusion.Graphs
Assembly: Vertex.js

Syntax

JavaScript Copy Code

```
function Vertex ()
```

See Also

Vertex Class
MindFusion.Graphs Namespace

Vertex Methods

Vertex.adjacentTo Method

Checks if there is an edge connecting this vertex with the specified vertex.

Namespace: MindFusion.Graphs
Assembly: Vertex.js

Syntax

JavaScript Copy Code

```
function adjacentTo (vertex)
```

Parameters

vertex
Vertex. The vertex to check.

Return Value

true if the vertices are connected; otherwise, **false**.

See Also

Vertex Members
Vertex Class
MindFusion.Graphs Namespace

Vertex.degree Method

Returns the number of incident edges.

Namespace: MindFusion.Graphs
Assembly: Vertex.js

Syntax

JavaScript Copy Code

```
function degree ()
```

Return Value

Number. The number of incident edges.

See Also

Vertex Members
Vertex Class
MindFusion.Graphs Namespace

Vertex.getCommonEdge Method

Finds the edge connecting the current vertex with the specified vertex.

Namespace: MindFusion.Graphs
Assembly: Vertex.js

Syntax

JavaScript Copy Code

```
function getCommonEdge (vertex)
```

Parameters

vertex
Vertex. The vertex to find an edge for.

Return Value

The edge connecting the vertices or **null**.

See Also

Vertex Members
Vertex Class
MindFusion.Graphs Namespace

Vertex.getNeighbors Method

Returns an array containing all vertices adjacent to the current vertex.

Namespace: MindFusion.Graphs
Assembly: Vertex.js

Syntax

JavaScript Copy Code

```
function getNeighbors ()
```

Return Value

Array. A list with all neighbors.

See Also

Vertex Members
Vertex Class
MindFusion.Graphs Namespace

Vertex.incidentWith Method

Checks if the current vertex is incident with the specified edge.

Namespace: MindFusion.Graphs
Assembly: Vertex.js

Syntax

JavaScript Copy Code

```
function incidentWith (edge)
```

Parameters

edge
Edge. The edge to check.

Return Value

true if the vertex is incident with the edge; otherwise, **false**.

See Also

Vertex Members
Vertex Class
MindFusion.Graphs Namespace

Server Side Reference

MindFusion.Diagramming Namespace

Action Enumeration

Defines values that identify the possible actions a user can carry out on diagram items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum Action
```

Visual Basic Copy Code

```
Public Enum Action
```

Members

	Member name	Description
	None	Do not perform any action.
	Create	Draw a new item.
	Modify	Modify an existing item.
	Split	Split a link's segment.

Remarks

Members of this enumeration are used as parameters of the InteractionState constructor.

See Also

MindFusion.Diagramming Namespace

AdjustmentHandle Class

Represents an adjustment handle used to move or modify diagram items.

Syntax

C# Copy Code

```
public class AdjustmentHandle
```

Visual Basic Copy Code

```
Public Class AdjustmentHandle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.AdjustmentHandle
    MindFusion.Diagramming.FreeFormHandle
    MindFusion.Diagramming.LinkAdjustmentHandle
    MindFusion.Diagramming.NodeAdjustmentHandle
    MindFusion.Diagramming.ShapeAdjustmentHandle

See Also

AdjustmentHandle Members
MindFusion.Diagramming Namespace

AdjustmentHandle Members

Represents an adjustment handle used to move or modify diagram items.

The following tables list the members exposed by the AdjustmentHandle type.

Public Constructors

	Name	Description
	AdjustmentHandle	Initializes a new instance of the **AdjustmentHandle** class.

Public Methods

	Name	Description
	IsMoveHandle	Returns a value indicating whether this handle is used to move items.
	IsResizeHandle	Gets whether this adjustment handle resizes the item.
	IsRotationHandle	Gets whether this adjustment handle changes item's rotation angle.

See Also

AdjustmentHandle Class
MindFusion.Diagramming Namespace

AdjustmentHandle Constructor

Initializes a new instance of the AdjustmentHandle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AdjustmentHandle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

AdjustmentHandle Class
MindFusion.Diagramming Namespace

AdjustmentHandle Methods

AdjustmentHandle.IsMoveHandle Method

Returns a value indicating whether this handle is used to move items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool IsMoveHandle ()
```

Visual Basic Copy Code

```
Public Overridable Function IsMoveHandle () As Boolean
```

Return Value

true if this is a move handle, or **false** otherwise.

See Also

AdjustmentHandle Members
AdjustmentHandle Class
MindFusion.Diagramming Namespace

AdjustmentHandle.IsResizeHandle Method

Gets whether this adjustment handle resizes the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool IsResizeHandle ()
```

Visual Basic Copy Code

```
Public Overridable Function IsResizeHandle () As Boolean
```

Return Value

true if this handle resizes the item, or false otherwise.

See Also

AdjustmentHandle Members
AdjustmentHandle Class
MindFusion.Diagramming Namespace

AdjustmentHandle.IsRotationHandle Method

Gets whether this adjustment handle changes item's rotation angle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool IsRotationHandle ()
```

Visual Basic Copy Code

```
Public Overridable Function IsRotationHandle () As Boolean
```

Return Value

true if this handle rotates the item, or false otherwise.

See Also

AdjustmentHandle Members
AdjustmentHandle Class
MindFusion.Diagramming Namespace

AdjustmentHandles Enumeration

Identifies the distinct selection handles of a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum AdjustmentHandles
```

Visual Basic Copy Code

```
Public Enum AdjustmentHandles
```

Members

	Member name	Description
	None	Specifies that none of the selection handles can be used to modify the node.
	ResizeTopLeft	The top-left handle is enabled and can be used to resize the node.
	ResizeTopRight	The top-right handle is enabled and can be used to resize the node.
	ResizeBottomRight	The bottom-right handle is enabled and can be used to resize the node.
	ResizeBottomLeft	The bottom-left handle is enabled and can be used to resize the node.

	ResizeTopCenter	The top-center handle is enabled and can be used to resize the node vertically.
	TopHandles	The top-left, top-center and top-right handles are enabled.
	ResizeMiddleRight	The middle-right handle is enabled and can be used to resize the node horizontally.
	RightHandles	The top-right, bottom-right and middle-right handles are enabled.
	ResizeBottomCenter	The bottom-center handle is enabled and can be used to resize the node vertically.
	BottomHandles	The bottom-left, bottom-center and bottom-right handles are enabled.
	ResizeMiddleLeft	The middle-left handle is enabled and can be used to resize the node horizontally.
	LeftHandles	The top-left, middle-left and bottom-left handles are enabled.
	Move	The center selection handle is enabled and can be used to move the node.
	Rotate	The rotation handle is enabled and can be used to rotate the node.
	All	All selection handles are enabled and can be used to modify the node.

Remarks

A combination of these flags can be assigned to the **EnabledHandles** property of nodes.

See Also

MindFusion.Diagramming Namespace

AeroEffect Class

Represents an effect that adds semitransparency to the node and emphasizes its outline by adding a shade and an inner stroke.

Syntax

C# Copy Code

```
public sealed class AeroEffect : NodeEffect
```

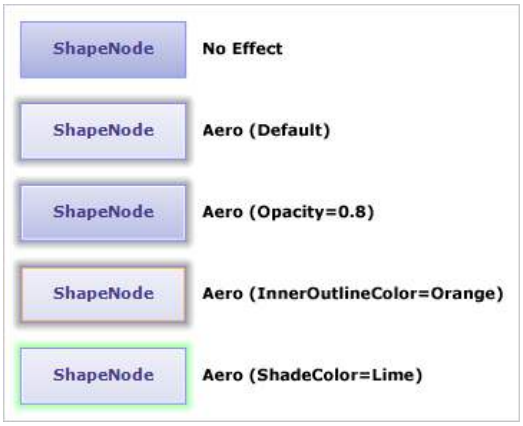
Visual Basic Copy Code

```
Public NotInheritable Class AeroEffect
    Inherits NodeEffect
```

Remarks

This effect can be applied to the nodes in a Diagram by adding an instance of this class to the NodeEffects collection.

The following image illustrates the Aero effect applied to a rectangular node as well as how the different properties affect the output:



Example

The following example demonstrates how to apply aero effect to the nodes of an existing diagram.

C# Copy Code

```
diagram.NodeEffects.Add(new AeroEffect());
```

Visual Basic Copy Code

```
diagram.NodeEffects.Add(New AeroEffect())
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.NodeEffect
    MindFusion.Diagramming.AeroEffect

See Also

AeroEffect Members
MindFusion.Diagramming Namespace

AeroEffect Members

Represents an effect that adds semitransparency to the node and emphasizes its outline by adding a shade and an inner stroke.

The following tables list the members exposed by the AeroEffect type.

Public Constructors

	Name	Description
	AeroEffect	Initializes a new instance of the **AeroEffect** class.

Public Properties

	Name	Description
	InnerOutlineColor	Gets or sets the color of the inner outline.
	Opacity	Gets or sets the opacity to apply to the node's background.
	ShadeColor	Gets or sets the color of the shade.

Public Methods

	Name	Description
	Clone	NodeEffect.Clone override.

Public Events

	Name	Description
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation. (Inherited from NodeEffect.)

See Also

AeroEffect Class
MindFusion.Diagramming Namespace

AeroEffect Constructor

Initializes a new instance of the AeroEffect class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AeroEffect ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

AeroEffect Class
MindFusion.Diagramming Namespace

AeroEffect Methods

AeroEffect.Clone Method

NodeEffect.Clone override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Object Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Object
```

Return Value

The copy of this effect.

See Also

AeroEffect Members
AeroEffect Class
MindFusion.Diagramming Namespace

AeroEffect Properties

AeroEffect.InnerOutlineColor Property

Gets or sets the color of the inner outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Color InnerOutlineColor { get; set; }
```

Visual Basic Copy Code

```
Public Property InnerOutlineColor As Color
```

Property Value

A **Microsoft.Maui.Graphic.Color** instance. The default is **White**.

See Also

AeroEffect Members
AeroEffect Class
MindFusion.Diagramming Namespace

AeroEffect.Opacity Property

Gets or sets the opacity to apply to the node's background.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Opacity { get; set; }
```

Visual Basic Copy Code

```
Public Property Opacity As Double
```

Property Value

A **double** value.

Remarks

Opacity varies in the range [0..1] where 0 indicates full transparency and 1 indicates the original transparency. Opacity has no effect on image backgrounds.

See Also

AeroEffect Members
AeroEffect Class
MindFusion.Diagramming Namespace

AeroEffect.ShadeColor Property

Gets or sets the color of the shade.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Color ShadeColor { get; set; }
```

Visual Basic Copy Code

```
Public Property ShadeColor As Color
```

Property Value

A **Microsoft.Maui.Graphic.Color** instance. The default is **Black**.

See Also

AeroEffect Members
AeroEffect Class
MindFusion.Diagramming Namespace

AeroEffect Events

Anchoring Enumeration

Defines values that specify how automatic layout algorithms align links to anchor points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum Anchoring
```

Visual Basic Copy Code

```
Public Enum Anchoring
```

Members

	Member name	Description
	Ignore	Anchor points are ignored.
	Keep	Links are aligned to their original anchor points.
	Reassign	Links are aligned to new anchor points, depending on the positions of graph nodes after a layout is applied.

Remarks

Assign values of this enumeration to the Anchoring property of layout algorithms. Most algorithms produce best results if they can place link ends at arbitrary positions over nodes' outlines, that is, when using the Ignore value. If your application logic requires links to be always attached to anchor points, use the other two values. In such cases Reassign would yield better results.

See Also

MindFusion.Diagramming Namespace
AnchorPattern Class
TreeLayout Class
GridLayout Class

LayeredLayout Class
SpringLayout Class

AnchorPattern Class

Represents a set of anchor points, which specify the exact locations where links are allowed to connect to nodes.

Syntax

C# Copy Code

```
public class AnchorPattern : ICloneable
```

Visual Basic Copy Code

```
Public Class AnchorPattern
    Implements ICloneable
```

Remarks

A pattern can be assigned as an integral entity to the AnchorPattern property of nodes. By default, when assigning a pattern to a node, a new copy of the pattern is created. Thus you can modify nodes' anchor patterns independently at runtime. However, if your anchor patterns won't change while the application executes, you can register them as shared to save storage space for the diagram. To do so, use AnchorPattern constructors that take a string id parameter. The **AnchorPattern** class exposes several predefined patterns as static properties; all of them are considered shared.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.AnchorPattern

See Also

AnchorPattern Members
MindFusion.Diagramming Namespace

AnchorPattern Members

Represents a set of anchor points, which specify the exact locations where links are allowed to connect to nodes.

The following tables list the members exposed by the AnchorPattern type.

Public Constructors

	Name	Description
	AnchorPattern	Overloaded. Initializes a new instance of the **AnchorPattern** class.

Public Properties

	Name	Description
	Decision1In3Out	Gets the predefined anchor pattern Decision1In3Out.
	Decision2In2Out	Gets the predefined anchor pattern Decision2In2Out.
	Id	Gets the **AnchorPattern** unique identifier.
	LeftInRightOut	Gets the predefined anchor pattern LeftInRightOut.
	Points	Gets a collection of the anchor points in this pattern.
	TopInBottomOut	Gets the predefined anchor pattern TopInBottomOut.

Public Methods

	Name	Description
	Clone	Creates an exact clone of this **AnchorPattern** object.
S	FromId	Returns a reference to the anchor pattern with the specified string identifier.
S	Register	Registers an anchor pattern as shared.

See Also

AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Constructor

Initializes a new instance of the AnchorPattern class.

Overload List

Name	Description
AnchorPattern ()	Initializes a new empty instance of the **AnchorPattern** class.
AnchorPattern (AnchorPoint[])	Initializes a new instance of the **AnchorPattern** class with the specified anchor points.
AnchorPattern (AnchorPoint[], String)	Initializes a new instance of the **AnchorPattern** class with the specified anchor points.
AnchorPattern (AnchorPointCollection)	Initializes a new instance of the **AnchorPattern** class with the specified anchor points.
AnchorPattern (AnchorPointCollection, String)	Initializes a new instance of the **AnchorPattern** class with the specified anchor points.
AnchorPattern (String)	Initializes a new empty instance of the **AnchorPattern** class.

See Also

AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Constructor ()

Initializes a new empty instance of the AnchorPattern class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

AnchorPattern Constructor Overload List
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Constructor (AnchorPoint[])

Initializes a new instance of the AnchorPattern class with the specified anchor points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern (
    AnchorPoint[] points
)
```

Visual Basic Copy Code

```
Public New ( _
    points() As AnchorPoint _
)
```

Parameters

points
Contains the initial set of AnchorPoint instances for the new pattern.

Example

The following code creates a decision node with two incoming anchors at the top and the left, and two outgoing at the bottom and the right vertices.

C# Copy Code

```
AnchorPattern decision2In2Out =
    new AnchorPattern(new AnchorPoint[] {
        new AnchorPoint(50, 0, true, false, MarkStyle.Circle, Color.Blue),
        new AnchorPoint(0, 50, true, false, MarkStyle.Circle, Color.Blue),
        new AnchorPoint(50, 100, false, true, MarkStyle.Rectangle, Color.Green),
        new AnchorPoint(100, 50, false, true, MarkStyle.Rectangle, Color.Red)
    });

ShapeNode decisionNode = diagram.Factory.CreateShapeNode(10, 10, 50, 35);
decisionNode.AnchorPattern = decision2In2Out;
```

Visual Basic Copy Code

```
Dim decision2In2Out As AnchorPattern = _
    New AnchorPattern(New AnchorPoint() { _
        New AnchorPoint(50, 0, True, False, MarkStyle.Circle, Color.Blue), _
        New AnchorPoint(0, 50, True, False, MarkStyle.Circle, Color.Blue), _
        New AnchorPoint(50, 100, False, True, MarkStyle.Rectangle, Color.Green), _
        New AnchorPoint(100, 50, False, True, MarkStyle.Rectangle, Color.Red) _
    })

Dim decisionNode As ShapeNode = Diagram.Factory.CreateShapeNode(10, 10, 50, 35)
decisionNode.AnchorPattern = decision2In2Out
```

See Also

AnchorPattern Constructor Overload List
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Constructor (AnchorPoint[], String)

Initializes a new instance of the AnchorPattern class with the specified anchor points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern (
    AnchorPoint[] points,
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    points() As AnchorPoint, _
    id As String _
)
```

Parameters

points

Contains the initial set of AnchorPoint instances for the new pattern.

id

An unique identifier for this anchor pattern.

See Also

AnchorPattern Constructor Overload List
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Constructor (AnchorPointCollection)

Initializes a new instance of the AnchorPattern class with the specified anchor points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern (
    AnchorPointCollection points
)
```

Visual Basic Copy Code

```
Public New ( _
    points As AnchorPointCollection _
)
```

Parameters

points

A collection of AnchorPoint objects.

See Also

AnchorPattern Constructor Overload List
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Constructor (AnchorPointCollection, String)

Initializes a new instance of the AnchorPattern class with the specified anchor points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern (
    AnchorPointCollection points,
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    points As AnchorPointCollection, _
    id As String _
)
```

Parameters

points

A collection of AnchorPoint objects.

id

An unique identifier for this anchor pattern.

See Also

AnchorPattern Constructor Overload List
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Constructor (String)

Initializes a new empty instance of the AnchorPattern class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern (
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String _
)
```

Parameters

id

An unique identifier for this anchor pattern.

See Also

AnchorPattern Constructor Overload List
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Methods

AnchorPattern.Clone Method

Creates an exact clone of this AnchorPattern object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As Object
```

Return Value

An instance of the AnchorPattern class.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern.FromId Method

Returns a reference to the anchor pattern with the specified string identifier.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static AnchorPattern FromId (
    string id
)
```

Visual Basic Copy Code

```
Public Shared Function FromId( _
    id As String _
) As AnchorPattern
```

Parameters

id

A string identifying an AnchorPattern instance.

Return Value

The **AnchorPattern** object whose Id is equal to the id parameter, or **null** (**Nothing** in Visual Basic) if no such object exists.

Remarks

The *id* argument should match an identifier passed to the constructor of an existing AnchorPattern object.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace
Shape Property
Shapes Property

AnchorPattern.Register Method

Registers an anchor pattern as shared.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static void Register (
    AnchorPattern pattern
)
```

Visual Basic Copy Code

```
Public Shared Sub Register( _
    pattern As AnchorPattern _
)
```

Parameters

pattern
The AnchorPattern to register.

Remarks

This method is obsolete. Please use the Id property to create shared anchor patterns.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern Properties

AnchorPattern.Decision1In3Out Property

Gets the predefined anchor pattern **Decision1In3Out**.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static AnchorPattern Decision1In3Out { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Decision1In3Out As AnchorPattern
```

Property Value

An instance of the AnchorPattern class.

Remarks

Incoming links connect to the top vertex; outgoing links can start from the other vertices.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern.Decision2In2Out Property

Gets the predefined anchor pattern **Decision2In2Out**.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static AnchorPattern Decision2In2Out { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Decision2In2Out As AnchorPattern
```

Property Value

An instance of the AnchorPattern class.

Remarks

Incoming links can connect with the top and left vertices; outgoing links can start from the right and bottom vertices.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern.Id Property

Gets the AnchorPattern unique identifier.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Id { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Id As String
```

Property Value

The string id that was passed as an argument to the AnchorPattern constructor. The default is an empty string.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern.LeftInRightOut Property

Gets the predefined anchor pattern **LeftInRightOut**.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static AnchorPattern LeftInRightOut { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property LeftInRightOut As AnchorPattern
```

Property Value

An instance of the AnchorPattern class.

Remarks

The middle point of a node's bounding rectangle left side is anchor for incoming links; that on the right side - for outgoing.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern.Points Property

Gets a collection of the anchor points in this pattern.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPointCollection Points { get; set; }
```

Visual Basic Copy Code

```
Public Property Points As AnchorPointCollection
```

Property Value

The string id that was passed as an argument to the AnchorPattern constructor. The default is an empty string.

Remarks

To find an anchor pattern by its id, call the static FromId method.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPattern.TopInBottomOut Property

Gets the predefined anchor pattern **TopInBottomOut**.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static AnchorPattern TopInBottomOut { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property TopInBottomOut As AnchorPattern
```

Property Value

An instance of the AnchorPattern class.

Remarks

The middle point of a node's bounding rectangle top side is anchor for incoming links; that on the bottom - for outgoing.

See Also

AnchorPattern Members
AnchorPattern Class
MindFusion.Diagramming Namespace

AnchorPoint Class

Specify the locations in a node, to which the user is allowed to connect links.

Syntax

C# Copy Code

```
public class AnchorPoint : ICloneable
```

Visual Basic Copy Code

```
Public Class AnchorPoint
    Implements ICloneable
```

Remarks

An anchor point can accept incoming or outgoing links, or both, as specified by the AllowIncoming and AllowOutgoing properties. The position of the anchor point inside a node's bounding rectangle is set through the X and Y properties, expressed as percentage of the rectangle width and height.

How anchor points are marked on the display can be set using MarkStyle and Color properties. Custom painting can be done by setting **MarkStyle** to Custom and handling the DrawAnchorPoint event.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.AnchorPoint

See Also

AnchorPoint Members
MindFusion.Diagramming Namespace

AnchorPoint Members

Specify the locations in a node, to which the user is allowed to connect links.

The following tables list the members exposed by the AnchorPoint type.

Public Constructors

	Name	Description

	AnchorPoint	Overloaded. The following constructors are provided for creating new **AnchorPoint** objects.

Public Properties

	Name	Description
	AllowIncoming	Gets or sets a value indicating whether incoming links can be connected to this anchor.
	AllowOutgoing	Gets or sets a value indicating whether the anchor point accepts outgoing connections.
	Color	Gets or sets the color of the anchor point mark.
	Column	Gets or sets the column index of a cell with which the anchor point is associated.
	MarkStyle	Gets or sets the appearance of the anchor point mark.
	Tag	Gets or sets a user-defined data associated with the anchor point.
	ToolTip	Gets or sets a tooltip text that should be displayed when the mouse hovers over an anchor point mark.
	X	Gets or sets the horizontal position of the anchor point expressed as percentage of a node's width.
	XUnit	Gets or sets whether the point's X position is specified as percentage of node's size or fixed offset from the node's left side.
	Y	Gets or sets the vertical position of the anchor point expressed as percentage of a node's height.
	YUnit	Gets or sets whether the point's Y position is specified as percentage of node's size or fixed offset from the node's top side.

Public Methods

	Name	Description
	Clone	Creates an exact clone of this **AnchorPoint** object.

Protected Methods

	Name	Description
	Draw	Renders the anchor point.
	LoadFromJson	Deserializes the anchor point from JSON format.
	LoadFromXml	Deserializes the anchor point from XML format.
	SaveToJson	Serializes the anchor point in JSON format.
	SaveToXml	Serializes the anchor point in XML format.

See Also

AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Constructor

The following constructors are provided for creating new AnchorPoint objects.

Overload List

Name	Description
AnchorPoint ()	Constructor called when deserializing **AnchorPoint** objects.
AnchorPoint (Double, Double)	Initializes a new instance of the **AnchorPoint** class with the specified coordinates.
AnchorPoint (Double, Double, Boolean, Boolean)	Initializes a new instance of the **AnchorPoint** class with the specified coordinates and restrictions.
AnchorPoint (Double, Double, Boolean, Boolean, Color, Int32)	Initializes a new instance of the **AnchorPoint** class with the specified coordinates, restrictions and color.
AnchorPoint (Double, Double, Boolean, Boolean, Color, Int32, Object)	Initializes a new instance of the **AnchorPoint** class with the specified coordinates, restrictions, color and tag.
AnchorPoint (Double, Double, Boolean, Boolean, MarkStyle)	Initializes a new instance of the **AnchorPoint** class with the specified coordinates, restrictions and style.
AnchorPoint (Double, Double, Boolean, Boolean, MarkStyle, Color)	Initializes a new instance of the **AnchorPoint** class with the specified coordinates, restrictions, style and color.

See Also

AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Constructor ()

Constructor called when deserializing AnchorPoint objects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPoint ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

AnchorPoint Constructor Overload List
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Constructor (Double, Double)

Initializes a new instance of the AnchorPoint class with the specified coordinates.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPoint (
    double x,
    double y
)
```

Visual Basic

```
Public New ( _
    x As Double, _
    y As Double _
)
```

Parameters

x
The horizontal position of the anchor expressed as percentage of the width of a node.

y
The vertical position of the anchor expressed as percentage of the height of a node.

See Also

AnchorPoint Constructor Overload List
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Constructor (Double, Double, Boolean, Boolean)

Initializes a new instance of the AnchorPoint class with the specified coordinates and restrictions.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public AnchorPoint (
    double x,
    double y,
    bool allowIncoming,
    bool allowOutgoing
)
```

Visual Basic

```
Public New ( _
    x As Double, _
    y As Double, _
    allowIncoming As Boolean, _
    allowOutgoing As Boolean _
)
```

Parameters

x
The horizontal position of the anchor expressed as percentage of the width of a node.

y
The vertical position of the anchor expressed as percentage of the height of a node.

allowIncoming
Specifies whether incoming links can connect to this anchor.

allowOutgoing
Specifies whether outgoing links can connect to this anchor.

See Also

AnchorPoint Constructor Overload List
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Constructor (Double, Double, Boolean, Boolean, Color, Int32)

Initializes a new instance of the AnchorPoint class with the specified coordinates, restrictions and color.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPoint (
    double x,
    double y,
    bool allowIncoming,
    bool allowOutgoing,
    Color color,
    int col
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Double, _
    y As Double, _
    allowIncoming As Boolean, _
    allowOutgoing As Boolean, _
    color As Color, _
    col As Integer _
)
```

Parameters

x
The horizontal position of the anchor expressed as percentage of the width of a node.

y
The vertical position of the anchor expressed as percentage of the height of a node.

allowIncoming
Specifies whether incoming links can connect to this anchor.

allowOutgoing
Specifies whether outgoing links can connect to this anchor.

color
The color with which the anchor mark is painted.

col
Specifies a table's column, with which an anchor point is associated.

See Also

AnchorPoint Constructor Overload List
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Constructor (Double, Double, Boolean, Boolean, Color, Int32, Object)

Initializes a new instance of the AnchorPoint class with the specified coordinates, restrictions, color and tag.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPoint (
    double x,
    double y,
    bool allowIncoming,
    bool allowOutgoing,
    Color color,
    int col,
    Object tag
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Double, _
    y As Double, _
    allowIncoming As Boolean, _
    allowOutgoing As Boolean, _
    color As Color, _
    col As Integer, _
    tag As Object _
)
```

Parameters

x
The horizontal position of the anchor expressed as percentage of the width of a node.

y
The vertical position of the anchor expressed as percentage of the height of a node.

allowIncoming
Specifies whether incoming links can connect to this anchor.

allowOutgoing
Specifies whether outgoing links can connect to this anchor.

color
The color with which the anchor mark is painted.

col
The table's column, with which an anchor point is associated.

tag
User-defined data associated with the anchor point.

See Also

AnchorPoint Constructor Overload List
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Constructor (Double, Double, Boolean, Boolean, MarkStyle)

Initializes a new instance of the AnchorPoint class with the specified coordinates, restrictions and style.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPoint (
    double x,
    double y,
    bool allowIncoming,
    bool allowOutgoing,
    MarkStyle markStyle
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Double, _
    y As Double, _
    allowIncoming As Boolean, _
    allowOutgoing As Boolean, _
    markStyle As MarkStyle _
)
```

Parameters

x
The horizontal position of the anchor expressed as percentage of the width of a node.

y
The vertical position of the anchor expressed as percentage of the height of a node.

allowIncoming
Specifies whether incoming links can connect to this anchor.

allowOutgoing
Specifies whether outgoing links can connect to this anchor.

markStyle
Defines how the anchor point is marked visually.

See Also

AnchorPoint Constructor Overload List
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Constructor (Double, Double, Boolean, Boolean, MarkStyle, Color)

Initializes a new instance of the AnchorPoint class with the specified coordinates, restrictions, style and color.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPoint (
    double x,
    double y,
    bool allowIncoming,
    bool allowOutgoing,
    MarkStyle markStyle,
    Color color
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Double, _
    y As Double, _
    allowIncoming As Boolean, _
    allowOutgoing As Boolean, _
    markStyle As MarkStyle, _
    color As Color _
)
```

Parameters

x
The horizontal position of the anchor expressed as percentage of the width of a node.

y
The vertical position of the anchor expressed as percentage of the height of a node.

allowIncoming
Specifies whether incoming links can connect to this anchor.

allowOutgoing
Specifies whether outgoing links can connect to this anchor.
markStyle
Defines how the anchor point is marked visually.
color
The color with which the anchor mark is painted.

See Also

AnchorPoint Constructor Overload List
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Methods

AnchorPoint.Clone Method

Creates an exact clone of this AnchorPoint object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Object Clone ()
```

Visual Basic Copy Code

```
Public Overridable Function Clone () As Object
```

Return Value

An instance of the AnchorPoint class.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.Draw Method

Renders the anchor point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void Draw (
    IGraphics graphics,
    Rect nodeRect,
    double angle,
    Point pivot,
    DrawAnchorPointEventArgs customDrawArgs,
    double markSize
)
```

Visual Basic Copy Code

```
Protected Overridable Sub Draw( _
    graphics As IGraphics, _
    nodeRect As Rect, _
    angle As Double, _
    pivot As Point, _
    customDrawArgs As DrawAnchorPointEventArgs, _
    markSize As Double _
)
```

Parameters

graphics

nodeRect

angle

pivot

customDrawArgs

markSize

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.LoadFromJson Method

Deserializes the anchor point from JSON format.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject instance containing the data to load.

context
A JsonPersistContext instance providing contextual information about the serialization process and helper serialization methods.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.LoadFromXml Method

Deserializes the anchor point from XML format.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

context

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.SaveToJson Method

Serializes the anchor point in JSON format.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject instance to contain the serialized data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.SaveToXml Method

Serializes the anchor point in XML format.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

context

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint Properties

AnchorPoint.AllowIncoming Property

Gets or sets a value indicating whether incoming links can be connected to this anchor.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowIncoming { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowIncoming As Boolean
```

Property Value

true to allow incoming links connect to the anchor; **false** to forbid that. The default is **true**.

Remarks

If no incoming anchors are defined for a node, links end points can be placed anywhere inside that node, unless AllowUnanchoredLinks is disabled.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AllowOutgoing Property
ValidateAnchorPoint Event

AnchorPoint.AllowOutgoing Property

Gets or sets a value indicating whether the anchor point accepts outgoing connections.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowOutgoing { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowOutgoing As Boolean
```

Property Value

true to allow outgoing links connect to the anchor; **false** to forbid that. The default is **true**.

Remarks

If no outgoing anchors are defined for a node, links start points can be placed anywhere inside that node, unless AllowUnanchoredLinks is disabled.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace
AllowIncoming Property
ValidateAnchorPoint Event

AnchorPoint.Color Property

Gets or sets the color of the anchor point mark.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Color Color { get; set; }
```

Visual Basic Copy Code

```
Public Property Color As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** instance. The default is **Color.Black**.

Remarks

The anchor mark shape specified by MarkStyle is painted with this color.

See Also

AnchorPoint Members
AnchorPoint Class

AnchorPoint.Column Property

Gets or sets the column index of a cell with which the anchor point is associated.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Column { get; set; }
```

Visual Basic Copy Code

```
Public Property Column As Integer
```

Property Value

An integer value specifying the zero-based index of a table's column. The default is -1.

Remarks

This property is useful when the anchor point is included in table row's AnchorPattern. It defines whether the anchor point is relative to a cell or to a whole row. If **Column** is set to -1, the point relative coordinates specify percents of row's extents. Otherwise the point coordinates specify percents of cell's extents.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace
Row.AnchorPattern Property

AnchorPoint.MarkStyle Property

Gets or sets the appearance of the anchor point mark.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MarkStyle MarkStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property MarkStyle As MarkStyle
```

Property Value

A member of the MarkStyle enumeration. The default is **MarkStyle.X**.

Remarks

Specifies how anchor points are marked visually on the display. Custom code can be used to draw the marks by setting the property value to MarkStyle.Custom and handling the DrawAnchorPoint event.

See Also

AnchorPoint Members
AnchorPoint Class

AnchorPoint.Tag Property

Gets or sets a user-defined data associated with the anchor point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Tag { get; set; }
```

Visual Basic Copy Code

```
Public Property Tag As Object
```

Property Value

An instance of any class derived from **System.Object**. The default is **null**.

Remarks

This property holds data that you define. MvcDiagram does not use it in any way. You might associate with the anchor point any value of primitive data types. The tag also might be a complex object from the .NET class library or a class defined by you. If the tag is an instance of value type, it will be serialized together with the object containing it when saving / loading a diagram. Reference types are serialized if marked with the **[Serializable]** attribute or if they implement the **ISerializable** interface.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace
SerializeTag Event

AnchorPoint.ToolTip Property

Gets or sets a tooltip text that should be displayed when the mouse hovers over an anchor point mark.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ToolTip { get; set; }
```

Visual Basic Copy Code

```
Public Property ToolTip As String
```

Property Value

A string specifying the tool-tip text displayed for this anchor point. The default is an empty string.

Remarks

When the mouse rests for a while over an anchor point, the tool-tip text is displayed in a small popup window near the mouse cursor. If the anchor point tooltip is an empty string, the ToolTip of the node to which the anchor point is assigned is displayed instead.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace

AnchorPoint.X Property

Gets or sets the horizontal position of the anchor point expressed as percentage of a node's width.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double X { get; set; }
```

Visual Basic Copy Code

```
Public Property X As Double
```

Property Value

A **double** value specifying the location of the anchor point, expressed as percentage.

Remarks

Set the property value to a number from 0 to 100, otherwise you might get unexpected behavior.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace
Y Property
XUnit Property

AnchorPoint.XUnit Property

Gets or sets whether the point's X position is specified as percentage of node's size or fixed offset from the node's left side.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public UnitType XUnit { get; set; }
```

Visual Basic Copy Code

```
Public Property XUnit As UnitType
```

Property Value

A member of the UnitType enumeration. The default value is **Percentage**.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace
X Property
YUnit Property

AnchorPoint.Y Property

Gets or sets the vertical position of the anchor point expressed as percentage of a node's height.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Y { get; set; }
```

Visual Basic Copy Code

```
Public Property Y As Double
```

Property Value

A **double** value specifying the location of the anchor point, expressed as percentage.

Remarks

Set the property value to a number from 0 to 100, otherwise you might get unexpected behavior.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace
X Property
YUnit Property

AnchorPoint.YUnit Property

Gets or sets whether the point's Y position is specified as percentage of node's size or fixed offset from the node's top side.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public UnitType YUnit { get; set; }
```

Visual Basic Copy Code

```
Public Property YUnit As UnitType
```

Property Value

A member of the UnitType enumeration. The default value is **Percentage**.

See Also

AnchorPoint Members
AnchorPoint Class
MindFusion.Diagramming Namespace
Y Property
XUnit Property

AnchorPointCollection Class

Represents a strongly typed collection of anchor points.

Syntax

C# Copy Code

```
public class AnchorPointCollection : CollectionBase<AnchorPoint>
```

Visual Basic Copy Code

```
Public Class AnchorPointCollection
    Inherits CollectionBase(Of AnchorPoint)
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.CollectionBase(AnchorPoint)
    MindFusion.Diagramming.AnchorPointCollection

See Also

AnchorPointCollection Members
MindFusion.Diagramming Namespace

AnchorPointCollection Members

Represents a strongly typed collection of anchor points.

The following tables list the members exposed by the AnchorPointCollection type.

Public Constructors

	Name	Description
	AnchorPointCollection	Overloaded.

Protected Fields

	Name	Description
	InnerList	Gets a **List<T>** containing the list of elements in the collection instance. (Inherited from CollectionBase.)

Public Properties

	Name	Description
	Count	Gets the number of elements actually contained in the collection. (Inherited from CollectionBase.)
	IsReadOnly	Gets a value indicating whether the collection is read-only. (Inherited from CollectionBase.)
	Item	Gets or sets the element at the specified index. (Inherited from CollectionBase.)

Public Methods

	Name	Description
	Add	Adds an object to the end of the collection. (Inherited from CollectionBase.)
	Add	Adds the specified anchor point to the collection.
	Clear	Removes all anchor points from the collection.

	Contains	Determines whether an element is in the collection. (Inherited from CollectionBase.)
	CopyTo	Copies the entire collection to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from CollectionBase.)
	GetAt	Gets the anchor point at the specified position.
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from CollectionBase.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire collection. (Inherited from CollectionBase.)
	Insert	Inserts an element into the collection at the specified index. (Inherited from CollectionBase.)
	Insert	Inserts the specified anchor point in the collection at the specified position.
	Remove	Removes the first occurrence of a specific object from the collection. (Inherited from CollectionBase.)
	Remove	Removes the specified anchor point from the collection.
	RemoveAt	Removes the anchor point at the specified index of the collection.
	Sort	Overloaded. (Inherited from CollectionBase.)

See Also

AnchorPointCollection Class
MindFusion.Diagramming Namespace

AnchorPointCollection Constructor

Overload List

Name	Description
AnchorPointCollection ()	Initializes a new instance of the AnchorPointCollection class.
AnchorPointCollection (AnchorPoint[])	Initializes a new instance of the **AnchorPointCollection** class.

See Also

AnchorPointCollection Class
MindFusion.Diagramming Namespace

AnchorPointCollection Constructor ()

Initializes a new instance of the AnchorPointCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPointCollection ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

AnchorPointCollection Constructor Overload List
AnchorPointCollection Class
MindFusion.Diagramming Namespace

AnchorPointCollection Constructor (AnchorPoint[])

Initializes a new instance of the AnchorPointCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPointCollection (
    AnchorPoint[] array
)
```

Visual Basic Copy Code

```
Public New ( _
    array() As AnchorPoint _
)
```

Parameters

array

See Also

AnchorPointCollection Constructor Overload List
AnchorPointCollection Class
MindFusion.Diagramming Namespace

AnchorPointCollection Methods

AnchorPointCollection.Add Method

Adds the specified anchor point to the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Add (
    AnchorPoint point
)
```

Visual Basic Copy Code

```
Public Overrides Sub Add( _
    point As AnchorPoint _
)
```

Parameters

point

See Also

AnchorPointCollection Members
AnchorPointCollection Class
MindFusion.Diagramming Namespace

AnchorPointCollection.Clear Method

Removes all anchor points from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Clear ()
```

Visual Basic Copy Code

```
Public Overrides Sub Clear ()
```

See Also

AnchorPointCollection Members
AnchorPointCollection Class
MindFusion.Diagramming Namespace

AnchorPointCollection.GetAt Method

Gets the anchor point at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPoint GetAt (
    int index
)
```

Visual Basic Copy Code

```
Public Function GetAt( _
    index As Integer _
) As AnchorPoint
```

Parameters

index

Return Value

See Also

AnchorPointCollection Members
AnchorPointCollection Class
MindFusion.Diagramming Namespace

AnchorPointCollection.Insert Method

Inserts the specified anchor point in the collection at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Insert (
    int index,
    AnchorPoint point
)
```

Visual Basic Copy Code

```
Public Overrides Sub Insert( _
    index As Integer, _
    point As AnchorPoint _
)
```

Parameters

index

point

See Also

AnchorPointCollection Members
AnchorPointCollection Class
MindFusion.Diagramming Namespace

AnchorPointCollection.Remove Method

Removes the specified anchor point from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool Remove (
    AnchorPoint point
)
```

Visual Basic Copy Code

```
Public Overrides Function Remove( _
    point As AnchorPoint _
) As Boolean
```

Parameters

point

Return Value

See Also

AnchorPointCollection Members
AnchorPointCollection Class
MindFusion.Diagramming Namespace

AnchorPointCollection.RemoveAt Method

Removes the anchor point at the specified index of the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void RemoveAt (
    int index
)
```

Visual Basic Copy Code

```
Public Overrides Sub RemoveAt( _
    index As Integer _
)
```

Parameters

index

See Also

AnchorPointCollection Members
AnchorPointCollection Class
MindFusion.Diagramming Namespace

AnchorPointEventArgs Class

Contains the arguments passed to the anchor point-related events, such as OriginAnchorChanged and DestinationAnchorChanged.

Syntax

C# Copy Code

```
public class AnchorPointEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class AnchorPointEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.AnchorPointEventArgs

See Also

AnchorPointEventArgs Members
MindFusion.Diagramming Namespace

AnchorPointEventArgs Members

Contains the arguments passed to the anchor point-related events, such as OriginAnchorChanged and DestinationAnchorChanged.

The following tables list the members exposed by the AnchorPointEventArgs type.

Public Constructors

	Name	Description
	AnchorPointEventArgs	Initializes a new instance of the **AnchorPointEventArgs** class.

Public Properties

	Name	Description
	Link	Gets the link whose anchor point index has been changed.
	NewAnchorIndex	Gets the index of the new anchor point.
	OldAnchorIndex	Gets the index of the old anchor point.

See Also

AnchorPointEventArgs Class
MindFusion.Diagramming Namespace

AnchorPointEventArgs Constructor

Initializes a new instance of the AnchorPointEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPointEventArgs (
    DiagramLink link,
    int oldAnchorIndex,
    int newAnchorIndex
)
```

Visual Basic Copy Code

```
Public New ( _
    link As DiagramLink, _
    oldAnchorIndex As Integer, _
    newAnchorIndex As Integer _
)
```

Parameters

link
The link whose end has been attached to a different anchor point.

oldAnchorIndex
The index of the old anchor point.

newAnchorIndex
The index of the new anchor point.

See Also

AnchorPointEventArgs Class
MindFusion.Diagramming Namespace

AnchorPointEventArgs Properties

AnchorPointEventArgs.Link Property

Gets the link whose anchor point index has been changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink Link { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Link As DiagramLink
```

Property Value

An instance of the DiagramLink class.

See Also

AnchorPointEventArgs Members
AnchorPointEventArgs Class
MindFusion.Diagramming Namespace

AnchorPointEventArgs.NewAnchorIndex Property

Gets the index of the new anchor point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int NewAnchorIndex { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NewAnchorIndex As Integer
```

Property Value

An integer value that represents the index of the anchor point within its containing AnchorPattern class.

See Also

AnchorPointEventArgs Members
AnchorPointEventArgs Class
MindFusion.Diagramming Namespace

AnchorPointEventArgs.OldAnchorIndex Property

Gets the index of the old anchor point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int OldAnchorIndex { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property OldAnchorIndex As Integer
```

Property Value

An integer value that represents the index of the anchor point within its containing AnchorPattern class.

See Also

AnchorPointEventArgs Members
AnchorPointEventArgs Class
MindFusion.Diagramming Namespace

ArcTemplate Class

Represents an arc in a shape definition.

Syntax

C# Copy Code

```
public class ArcTemplate : ElementTemplate
```

Visual Basic Copy Code

```
Public Class ArcTemplate
    Inherits ElementTemplate
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ElementTemplate
    MindFusion.Diagramming.ArcTemplate

See Also

ArcTemplate Members
MindFusion.Diagramming Namespace

ArcTemplate Members

Represents an arc in a shape definition.

The following tables list the members exposed by the ArcTemplate type.

Public Constructors

	Name	Description
	ArcTemplate	Overloaded. Initializes a new instance of the **ArcTemplate** class.

Public Properties

	Name	Description
	Bounds	Gets the bounding rectangle of this element.
	Color	Gets or sets the outline color of this element. (Inherited from ElementTemplate.)
	DashStyle	Gets or sets the dash style of the outline of this element. (Inherited from ElementTemplate.)
	StartAngle	Gets the start angle of this arc element.
	SweepAngle	Gets the sweep angle of this arc element.
	Width	Gets or sets the outline width of this element. (Inherited from ElementTemplate.)

Public Methods

	Name	Description
	Clone	Creates an exact copy of this **ArcTemplate** element.
	SameAs	ElementTemplate.SameAs override.

Protected Methods

	Name	Description
	GetElementPen	Gets a pen used to render the element. If no pen properties are explicitly specified for this element, returns the pen passed as an argument. (Inherited from ElementTemplate.)

See Also

ArcTemplate Class
MindFusion.Diagramming Namespace

ArcTemplate Constructor

Initializes a new instance of the ArcTemplate class.

Overload List

Name	Description
ArcTemplate (Double, Double, Double, Double, Double, Double)	Initializes a new instance of the **ArcTemplate** class with the specified coordinates.
ArcTemplate (Double, Double, Double, Double, Double, Double, Color, DashStyle, Double)	Initializes a new instance of the **ArcTemplate** class with the specified coordinates and visualization settings.

See Also

ArcTemplate Class
MindFusion.Diagramming Namespace

ArcTemplate Constructor (Double, Double, Double, Double, Double, Double)

Initializes a new instance of the ArcTemplate class with the specified coordinates.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ArcTemplate (
    double x,
    double y,
    double w,
    double h,
    double a,
    double s
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Double, _
    y As Double, _
    w As Double, _
    h As Double, _
    a As Double, _
    s As Double _
)
```

Parameters

x
The x-coordinate of the upper left corner of the ellipse's bounding rectangle.

y
The y-coordinate of the upper left corner of the ellipse's bounding rectangle.

w
The width of the ellipse's bounding rectangle.

h
The height of the ellipse's bounding rectangle.

a
The clockwise angle, in degrees, between the horizontal axis of the ellipse and the starting point of the arc.

s
The clockwise angle, in degrees, between the starting point and the ending point of the arc.

Remarks

An arc template is defined by an ellipse and starting and sweeping angles.

See Also

ArcTemplate Constructor Overload List
ArcTemplate Class
MindFusion.Diagramming Namespace

ArcTemplate Constructor (Double, Double, Double, Double, Double, Double, Color, DashStyle, Double)

Initializes a new instance of the ArcTemplate class with the specified coordinates and visualization settings.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ArcTemplate (
    double x,
    double y,
    double w,
    double h,
    double a,
    double s,
    Color color,
    DashStyle dashStyle,
    double width
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Double, _
    y As Double, _
    w As Double, _
    h As Double, _
    a As Double, _
    s As Double, _
    color As Color, _
```

```
    dashStyle As DashStyle, _
    width As Double _
)
```

Parameters

x
The x-coordinate of the upper left corner of the ellipse's bounding rectangle.

y
The y-coordinate of the upper left corner of the ellipse's bounding rectangle.

w
The width of the ellipse's bounding rectangle.

h
The height of the ellipse's bounding rectangle.

a
The clockwise angle, in degrees, between the horizontal axis of the ellipse and the starting point of the arc.

s
The clockwise angle, in degrees, between the starting point and the ending point of the arc.

color
The color of this segment.

dashStyle
The dash style of this segment.

width
The width of this segment.

Remarks

An arc template is defined by an ellipse and starting and sweeping angles.

See Also

ArcTemplate Constructor Overload List
ArcTemplate Class
MindFusion.Diagramming Namespace

ArcTemplate Methods

ArcTemplate.Clone Method

Creates an exact copy of this ArcTemplate element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Object Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Object
```

Return Value

An instance of the ArcTemplate class.

See Also

ArcTemplate Members
ArcTemplate Class
MindFusion.Diagramming Namespace

ArcTemplate.SameAs Method

ElementTemplate.SameAs override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool SameAs (
    ElementTemplate obj
)
```

Visual Basic Copy Code

```
Public Overrides Function SameAs( _
    obj As ElementTemplate _
) As Boolean
```

Parameters

obj
The ElementTemplate to compare to this template.

Return Value

true if the specified object is an ArcTemplate closely approximating this template; otherwise, **false**.

See Also

ArcTemplate Members
ArcTemplate Class
MindFusion.Diagramming Namespace

ArcTemplate Properties

ArcTemplate.Bounds Property

Gets the bounding rectangle of this element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect Bounds { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Bounds As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** instance.

See Also

ArcTemplate Members
ArcTemplate Class
MindFusion.Diagramming Namespace

ArcTemplate.StartAngle Property

Gets the start angle of this arc element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double StartAngle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property StartAngle As Double
```

Property Value

A **double** value specifying the start angle of the arc, in degrees.

See Also

ArcTemplate Members
ArcTemplate Class
MindFusion.Diagramming Namespace

ArcTemplate.SweepAngle Property

Gets the sweep angle of this arc element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double SweepAngle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property SweepAngle As Double
```

Property Value

A **double** value specifying the sweep angle of the arc, in degrees.

See Also

ArcTemplate Members
ArcTemplate Class
MindFusion.Diagramming Namespace

AreaEventArgs Class

Contains arguments passed to handlers of the CreatingArea event of the HtmlBuilder class.

Syntax

C# Copy Code

```
public class AreaEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class AreaEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.AreaEventArgs

See Also

AreaEventArgs Members
MindFusion.Diagramming Namespace

AreaEventArgs Members

Contains arguments passed to handlers of the CreatingArea event of the HtmlBuilder class.

The following tables list the members exposed by the AreaEventArgs type.

Public Properties

	Name	Description
	AreaTag	Gets or sets the HTML AREA tag created by **HtmlBuilder**. You can insert additional attributes or JavaScript event handlers into that string.
	Column	Gets the index of a column if the event is raised for a table cell.
	Item	Gets the diagram element represented by **AreaTag** in the image map.
	Row	Gets the index of a row if the event is raised for a table cell.

See Also

AreaEventArgs Class
MindFusion.Diagramming Namespace

AreaEventArgs Properties

AreaEventArgs.AreaTag Property

Gets or sets the HTML AREA tag created by HtmlBuilder. You can insert additional attributes or JavaScript event handlers into that string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string AreaTag { get; set; }
```

Visual Basic Copy Code

```
Public Property AreaTag As String
```

Property Value

A string value that represents the AREA tag, including the opening and closing brackets.

See Also

AreaEventArgs Members
AreaEventArgs Class
MindFusion.Diagramming Namespace

AreaEventArgs.Column Property

Gets the index of a column if the event is raised for a table cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Column { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Column As Integer
```

Property Value

An integer value specifying the zero-based index of the related column.

See Also

AreaEventArgs Members
AreaEventArgs Class
MindFusion.Diagramming Namespace

AreaEventArgs.Item Property

Gets the diagram element represented by AreaTag in the image map.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem Item { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Item As DiagramItem
```

Property Value

An instance of the DiagramItem class.

See Also

AreaEventArgs Members
AreaEventArgs Class
MindFusion.Diagramming Namespace

AreaEventArgs.Row Property

Gets the index of a row if the event is raised for a table cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Row { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Row As Integer
```

Property Value

An integer value specifying the zero-based index of the related row.

See Also

AreaEventArgs Members
AreaEventArgs Class
MindFusion.Diagramming Namespace

ArrowHeads Class

Contains predefined arrowhead shapes.

Syntax

C# Copy Code

```
public sealed class ArrowHeads
```

Visual Basic Copy Code

```
Public NotInheritable Class ArrowHeads
```

Remarks

Shape objects exposed as properties of this class can be assigned to the HeadShape, BaseShape and IntermediateShape properties of links.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ArrowHeads

See Also

ArrowHeads Members
MindFusion.Diagramming Namespace

ArrowHeads Members

Contains predefined arrowhead shapes.

The following tables list the members exposed by the ArrowHeads type.

Public Fields

	Name	Description
	Arrow	Display an arrow-like shape.
	BackSlash	Display a back slash.
	BowArrow	Display a bow arrow.

	Circle	Display a circle.
	DefaultFlow	"Default flow" in BPMN diagrams.
	DoubleArrow	Display a double arrow.
	None	Do not display an arrowhead shape.
	Pentagon	Display a pentagon.
	PointerArrow	Display a pointer arrow.
	Quill	Display a quill.
	Reversed	Reversed arrowhead.
	RevTriangle	Display a reversed triangle.
	RevWithCirc	Reversed arrowhead with a circle.
	RevWithLine	Reversed arrowhead with intersecting line.
	Rhombus	Display a rhombus.
	Slash	Display a slash.
	Tetragon	Display a tetragon.
	Triangle	Display a triangle.

See Also

ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads Fields

ArrowHeads.Arrow Field

Display an arrow-like shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Arrow
```

Visual Basic Copy Code

```
Public Shared Arrow As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.BackSlash Field

Display a back slash.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BackSlash
```

Visual Basic Copy Code

```
Public Shared BackSlash As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.BowArrow Field

Display a bow arrow.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BowArrow
```

Visual Basic Copy Code

```
Public Shared BowArrow As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Circle Field

Display a circle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Circle
```

Visual Basic Copy Code

```
Public Shared Circle As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.DefaultFlow Field

"Default flow" in BPMN diagrams.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape DefaultFlow
```

Visual Basic Copy Code

```
Public Shared DefaultFlow As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.DoubleArrow Field

Display a double arrow.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape DoubleArrow
```

Visual Basic Copy Code

```
Public Shared DoubleArrow As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.None Field

Do not display an arrowhead shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape None
```

Visual Basic Copy Code

```
Public Shared None As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Pentagon Field

Display a pentagon.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Pentagon
```

Visual Basic Copy Code

```
Public Shared Pentagon As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.PointerArrow Field

Display a pointer arrow.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape PointerArrow
```

Visual Basic Copy Code

```
Public Shared PointerArrow As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Quill Field

Display a quill.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Quill
```

Visual Basic Copy Code

```
Public Shared Quill As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Reversed Field

Reversed arrowhead.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Reversed
```

Visual Basic Copy Code

```
Public Shared Reversed As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.RevTriangle Field

Display a reversed triangle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape RevTriangle
```

Visual Basic Copy Code

```
Public Shared RevTriangle As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.RevWithCirc Field

Reversed arrowhead with a circle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape RevWithCirc
```

Visual Basic Copy Code

```
Public Shared RevWithCirc As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.RevWithLine Field

Reversed arrowhead with intersecting line.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape RevWithLine
```

Visual Basic Copy Code

```
Public Shared RevWithLine As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Rhombus Field

Display a rhombus.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Rhombus
```

Visual Basic Copy Code

```
Public Shared Rhombus As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Slash Field

Display a slash.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Slash
```

Visual Basic Copy Code

```
Public Shared Slash As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Tetragon Field

Display a tetragon.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Tetragon
```

Visual Basic Copy Code

```
Public Shared Tetragon As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

ArrowHeads.Triangle Field

Display a triangle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Triangle
```

Visual Basic Copy Code

```
Public Shared Triangle As Shape
```

See Also

ArrowHeads Members
ArrowHeads Class
MindFusion.Diagramming Namespace

AttachTo Enumeration

Specifies the type of attachment in a group.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum AttachTo
```

Visual Basic Copy Code

```
Public Enum AttachTo
```

Members

	Member name	Description
	FixedCorner	Indicates attachment to a fixed corner of a node.
	Proportional	Indicates proportional attachment to a node.
	LinkPoint	Indicates attachment to a link's control point.
	LinkSegment	Indicates attachment to a link's segment.
	LongestHSegment	Indicates attachment to the longest horizontal segment of a link.
	SideMiddle	Indicates attachment to the middle of a side of a node's bounding rectangle.
	AnchorToEdges	Indicates attachment that is anchored to one or more sides of a node.
	ControlledByMaster	The virtual UpdateGroup method of master items is called to update subordinates.
	DistanceAlongLink	Preserves the absolute distance from a node to link's start or end point.
	ProportionalToLength	Preserves the relative distance from a node to link's start point.
	FractionalPosition	Keeps attached node's size and maintains its original distance to point specified as fraction of main node's size.

See Also

MindFusion.Diagramming Namespace

AttachToLink Enumeration

Defines values specifying the possible ways to attach a node to a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum AttachToLink
```

Visual Basic — Copy Code

```
Public Enum AttachToLink
```

Members

	Member name	Description
	Point	The node is attached to a fixed control point of the link.
	Segment	The node is attached to a fixed segment of the link.
	LongestHSegment	The node stays attached to the longest horizontal segment of the link.
	FixedDistance	Preserves the absolute distance from a node to link's start or end point.
	RelativeDistance	Preserves the relative distance from a node to link's start point.

Remarks

Members of this enumeration are used as values for the *attType* argument of the AttachTo method of the DiagramNode class.

See Also

MindFusion.Diagramming Namespace

AttachToNode Enumeration

Defines values specifying the possible ways to attach a node to another node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# — Copy Code

```
public enum AttachToNode
```

Visual Basic — Copy Code

```
Public Enum AttachToNode
```

Members

	Member name	Description
	TopLeft	The node follows the top-left corner of the master node.
	TopRight	The node follows the top-right corner of the master node.
	BottomLeft	The node follows the bottom-left corner of the master node.
	BottomRight	The node follows the bottom-right corner of the master node.
	TopCenter	The node follows the top-center point of the master node.
	MiddleRight	The node follows the middle-right point of the master node.
	BottomCenter	The node follows the bottom-center point of the master node.
	MiddleLeft	The node follows the middle-left point of the master node.

Remarks

Members of this enumeration are used as values for the *attType* argument of the AttachTo method of the DiagramNode class.

See Also

MindFusion.Diagramming Namespace

AutoJson Class

Apply the **AutoJson** attribute to a property of a CompositeNode-derived class to indicate that it should be included in JSON serialization.

Syntax

C# Copy Code

```
public class AutoJson : Attribute
```

Visual Basic Copy Code

```
Public Class AutoJson
    Inherits Attribute
```

Inheritance Hierarchy

System.Object
  System.Attribute
    MindFusion.Diagramming.AutoJson

See Also

AutoJson Members
MindFusion.Diagramming Namespace

AutoJson Members

Apply the AutoJson attribute to a property of a CompositeNode-derived class to indicate that it should be included in JSON serialization.

The following tables list the members exposed by the **AutoJson** type.

Public Constructors

	Name	Description
	AutoJson	Initializes a new instance of the **AutoJson** attribute.

See Also

AutoJson Class
MindFusion.Diagramming Namespace

AutoJson Constructor

Initializes a new instance of the AutoJson attribute.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AutoJson ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

AutoJson Class
MindFusion.Diagramming Namespace

AutoResize Enumeration

Defines in which directions automatic resizing can enlarge the diagram scrollable area.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum AutoResize
```

Visual Basic Copy Code

```
Public Enum AutoResize
```

Members

	Member name	Description
	None	Do not enlarge the diagram area automatically.
	RightAndDown	Enlarge the diagram area to the right and down if an item is moved outside the boundaries in these directions.
	AllDirections	Enlarge the diagram area in any direction if an item is moved outside the boundaries.

Remarks

Values from this enumeration can be assigned to the AutoResize property.

See Also

MindFusion.Diagramming Namespace

Axis Enumeration

Specifies a coordinate axis.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum Axis
```

Visual Basic Copy Code

```
Public Enum Axis
```

Members

	Member name	Description
	X	Indicates the X coordinate axis.
	Y	Indicates the Y coordinate axis.

See Also

MindFusion.Diagramming Namespace

Behavior Enumeration

Defines values that specify how the component responds to actions performed by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum Behavior
```

Visual Basic Copy Code

```
Public Enum Behavior
```

Members

	Member name	Description
	Modify	Items can be selected and modified. New items cannot be created.
	DrawShapes	Drawing with the mouse creates ShapeNode instances.
	DrawLinks	Drawing started over a node creates a DiagramLink. Otherwise a lasso rectangle is displayed allowing the selection of items.
	LinkShapes	A mode that is suitable for creating flowcharts and process diagrams. Drawing over empty document area creates a **ShapeNode** instance. Drawing started over a node creates a **DiagramLink**.
	LinkTables	A mode that is suitable for creating entity-relationship diagrams. Drawing over empty document area creates a TableNode instance. Drawing started over a node creates a **DiagramLink**.
	DrawTables	Drawing with the mouse creates **TableNode** instances.
	DrawControls	Drawing with the mouse creates **ControlNode** instances.
	LinkControls	Drawing over empty document area creates a **ControlNode** instance. Drawing started over a node creates a **DiagramLink**.
	DoNothing	The control ignores users actions with the mouse, but raises the appropriate mouse events. This mode allows applications to implement their own mouse-drawing behavior, disabling the MvcDiagram default response to users actions.
	Custom	Specifies that user input will be handled by the **BehaviorBase**-derived object assigned to **CustomBehavior**.
	DrawContainers	Drawing with the mouse creates ContainerNode instances.
	LinkContainers	Drawing over empty document area creates a **ContainerNode** instance. Drawing started over a node creates a **DiagramLink**.

	Pan	Dragging with the mouse pans the view.
	DrawTreeViews	Drawing with the mouse creates TreeViewNode instances.
	LinkTreeViews	Drawing over empty document area creates a **TreeViewNode** instance. Drawing started over a node creates a **DiagramLink**.
	DrawSvgNodes	Drawing with the mouse creates SvgNode instances.
	LinkSvgNodes	Drawing over empty document area creates an **SvgNode** instance. Drawing started over a node creates a **DiagramLink**.
	SelectOnly	Allow only selection of existing items. Modifying them or drawing new ones is disabled.
	Magnify	Dragging the mouse magnifies the view. Interactive creation or modification of items is disabled.
	DrawVisioNodes	Drawing with the mouse creates **VisioNode** instances.
	LinkVisioNodes	Drawing over empty document area creates a **VisioNode** instance. Drawing started over a node creates a **DiagramLink**.
	DrawBarcodes	Drawing with the mouse creates **BarcodeNode** instances.
	LinkBarcodes	Drawing over empty document area creates a **BarcodeNode** instance. Drawing started over a node creates a **DiagramLink**.
	DrawFreeForms	Drawing with the mouse creates FreeFormNode instances.
	LinkFreeForms	Drawing over empty document area creates a **FreeFormNode** instance. Drawing started over a node creates a **DiagramLink**.
	DrawFreeShapes	Points from user input are shown using a **FreeFormNode** instance. Once the user finishes drawing, the **FreeFormNode** is replaced by a **ShapeNode** with a matching shape from Diagram.FreeFormTargets list.
	LinkFreeShapes	Points from user input are shown using a **FreeFormNode** instance. Once the user finishes drawing, the **FreeFormNode** is replaced by a **ShapeNode** with a matching shape from **Diagram.FreeFormTargets** list. Drawing started over a node creates a **DiagramLink**.
	PanAndModify	Items can be selected and modified. New items cannot be created, and dragging without grabbing an adjustment handle pans the view.
	MoveNodes	Dragging with the mouse moves the node under mouse pointer.
	Zoom	Dragging with the mouse draws a lasso tool that zooms the diagram view to selected rectangular area.
	MoveLabels	Dragging with the mouse moves the label under mouse pointer.
	DrawPagedContainers	Drawing with the mouse creates PagedContainerNode instances.
	LinkPagedContainers	Drawing over empty document area creates a PagedContainerNode instance. Drawing started over a node creates a DiagramLink.

Remarks

Set the **Behavior** property to specify how MvcDiagram interprets user's actions. In all modes:

- single clicking selects the item under the mouse pointer and make it the ActiveItem;
- dragging while CTRL is pressed creates multiple selection;
- dragging while ALT is pressed pans the view;

See Also

MindFusion.Diagramming Namespace
Behavior Property

BezierTemplate Class

Represents a Bézier curve in a shape definition.

Syntax

C# Copy Code

```
public class BezierTemplate : ElementTemplate
```

Visual Basic Copy Code

```
Public Class BezierTemplate
    Inherits ElementTemplate
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ElementTemplate
    MindFusion.Diagramming.BezierTemplate

See Also

BezierTemplate Members
MindFusion.Diagramming Namespace

BezierTemplate Members

Represents a Bézier curve in a shape definition.

The following tables list the members exposed by the BezierTemplate type.

Public Constructors

	Name	Description
	BezierTemplate	Overloaded. Initializes a new instance of the **BezierTemplate** class.

Public Properties

	Name	Description
	Color	Gets or sets the outline color of this element. (Inherited from ElementTemplate.)
	Coordinates	Gets an array containing the coordinates of the points defining the Bézier curve represented by this element.
	DashStyle	Gets or sets the dash style of the outline of this element. (Inherited from ElementTemplate.)
	Width	Gets or sets the outline width of this element. (Inherited from ElementTemplate.)

Public Methods

	Name	Description
	Clone	Creates an exact copy of this **BezierTemplate** element.
	SameAs	ElementTemplate.SameAs override.

Protected Methods

	Name	Description

	GetElementPen	Gets a pen used to render the element. If no pen properties are explicitly specified for this element, returns the pen passed as an argument. (Inherited from ElementTemplate.)

See Also

BezierTemplate Class
MindFusion.Diagramming Namespace

BezierTemplate Constructor

Initializes a new instance of the BezierTemplate class.

Overload List

Name	Description
BezierTemplate (Double, Double, Double, Double, Double, Double, Double, Double)	Initializes a new instance of the **BezierTemplate** class with the specified coordinates.
BezierTemplate (Double, Double, Double, Double, Double, Double, Color, DashStyle, Double)	Initializes a new instance of the **BezierTemplate** class with the specified coordinates and visualization settings.

See Also

BezierTemplate Class
MindFusion.Diagramming Namespace

BezierTemplate Constructor (Double, Double, Double, Double, Double, Double, Double, Double)

Initializes a new instance of the BezierTemplate class with the specified coordinates.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public BezierTemplate (
    double x1,
    double y1,
    double x2,
    double y2,
    double x3,
    double y3,
    double x4,
    double y4
)
```

Visual Basic Copy Code

```
Public New ( _
    x1 As Double, _
    y1 As Double, _
    x2 As Double, _
    y2 As Double, _
    x3 As Double, _
    y3 As Double, _
    x4 As Double, _
    y4 As Double _
)
```

Parameters

x1
The x-coordinate of the first control point of the curve.

y1
The y-coordinate of the first control point of the curve.

x2
The x-coordinate of the second control point of the curve.

y2
The y-coordinate of the second control point of the curve.

x3
The x-coordinate of the third control point of the curve.

y3
The y-coordinate of the third control point of the curve.

x4
The x-coordinate of the fourth control point of the curve.

y4
The y-coordinate of the fourth control point of the curve.

Remarks

A Bézier spline is defined by four control points. The spline goes through the first and last points, and the other two define the tension of the spline.

See Also

BezierTemplate Constructor Overload List
BezierTemplate Class
MindFusion.Diagramming Namespace

BezierTemplate Constructor (Double, Double, Double, Double, Double, Double, Double, Double, Color, DashStyle, Double)

Initializes a new instance of the BezierTemplate class with the specified coordinates and visualization settings.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public BezierTemplate (
    double x1,
    double y1,
    double x2,
    double y2,
    double x3,
    double y3,
    double x4,
    double y4,
    Color color,
    DashStyle dashStyle,
    double width
)
```

Visual Basic Copy Code

```
Public New ( _
    x1 As Double, _
    y1 As Double, _
    x2 As Double, _
    y2 As Double, _
    x3 As Double, _
    y3 As Double, _
    x4 As Double, _
    y4 As Double, _
```

```
    color As Color, _
    dashStyle As DashStyle, _
    width As Double _
)
```

Parameters

x1
The x-coordinate of the first control point of the curve.

y1
The y-coordinate of the first control point of the curve.

x2
The x-coordinate of the second control point of the curve.

y2
The y-coordinate of the second control point of the curve.

x3
The x-coordinate of the third control point of the curve.

y3
The y-coordinate of the third control point of the curve.

x4
The x-coordinate of the fourth control point of the curve

y4
The y-coordinate of the fourth control point of the curve.

color
The color of this segment.

dashStyle
The dash style of this segment.

width
The width of this segment.

Remarks

A Bézier spline is defined by four control points. The spline goes through the first and last points, and the other two define the tension of the spline.

See Also

BezierTemplate Constructor Overload List
BezierTemplate Class
MindFusion.Diagramming Namespace

BezierTemplate Methods

BezierTemplate.Clone Method

Creates an exact copy of this BezierTemplate element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Object Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Object
```

Return Value

An instance of the BezierTemplate class.

See Also

BezierTemplate Members
BezierTemplate Class
MindFusion.Diagramming Namespace

BezierTemplate.SameAs Method

ElementTemplate.SameAs override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool SameAs (
    ElementTemplate obj
)
```

Visual Basic Copy Code

```
Public Overrides Function SameAs( _
    obj As ElementTemplate _
) As Boolean
```

Parameters

obj
The ElementTemplate to compare to this template.

Return Value

true if the specified object is a BezierTemplate closely approximating this template; otherwise, **false**.

See Also

BezierTemplate Members
BezierTemplate Class
MindFusion.Diagramming Namespace

BezierTemplate Properties

BezierTemplate.Coordinates Property

Gets an array containing the coordinates of the points defining the Bézier curve represented by this element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double[] Coordinates { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Coordinates As Double()
```

Property Value

An array of **double** values containing the coordinates of the points composing this element.

See Also

BezierTemplate Members
BezierTemplate Class
MindFusion.Diagramming Namespace

BranchIndicator Enumeration

Defines values that indicate how to mark link branching points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum BranchIndicator
```

Visual Basic Copy Code

```
Public Enum BranchIndicator
```

Members

	Member name	Description
	None	Do not draw indicators.
	Circle	Draw a circle centered on the branching point.
	Arrow	Draw a bent arrow that also shows the link direction.
	Custom	Raise DrawBranchIndicator event to enable custom drawing of branch indicators.

See Also

MindFusion.Diagramming Namespace

CellEventArgs Class

Contains the arguments passed to event handlers, such as CellClicked.

Syntax

C# Copy Code

```
public class CellEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class CellEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.CellEventArgs
      MindFusion.Diagramming.CellValidationEventArgs

See Also

CellEventArgs Members

Contains the arguments passed to event handlers, such as CellClicked.

The following tables list the members exposed by the CellEventArgs type.

Public Constructors

	Name	Description
	CellEventArgs	Initializes a new instance of the **CellEventArgs** class.

Public Properties

	Name	Description
	Cell	Get the cell for which the event is raised.
	Column	Gets the index of the table's column that contains **Cell**.
	MouseButton	Gets which mouse button has been pressed.
	MousePosition	Gets the current mouse pointer position.
	PreventNodeEvent	Gets or sets whether to prevent node event from being raised after cell's one.
	Row	Gets the index of the table's row that contains **Cell**.
	Table	Get the table that contains **Cell**.

See Also

CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs Constructor

Initializes a new instance of the CellEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CellEventArgs (
    TableNode.Cell cell,
    MouseButton mouseButton,
    Point mousePosition,
    int column,
    int row
)
```

Visual Basic Copy Code

```
Public New ( _
    cell As TableNode.Cell, _
    mouseButton As MouseButton, _
    mousePosition As Point, _
    column As Integer, _
```

```
    row As Integer _
)
```

Parameters

cell
The TableNode.Cell related to the event.

mouseButton
The mouse button related to the event.

mousePosition
The position of the mouse at the time the event was raised.

column
The zero-based index of the column related to the event.

row
The zero-based index of the row related to the event.

See Also

CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs Properties

CellEventArgs.Cell Property

Get the cell for which the event is raised.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Cell Cell { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Cell As TableNode.Cell
```

Property Value

An instance of the TableNode.Cell class specifying the cell related to the event.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.Column Property

Gets the index of the table's column that contains Cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Column { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Column As Integer
```

Property Value

An integer value specifying the zero-based index of the table column related to the event.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.MouseButton Property

Gets which mouse button has been pressed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MouseButton MouseButton { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MouseButton As MouseButton
```

Property Value

A member of the MouseButton enumeration specifying the currently pressed mouse button.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.MousePosition Property

Gets the current mouse pointer position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point MousePosition { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MousePosition As Point
```

Property Value

A **Microsoft.Maui.Graphics.Point** specifying the current mouse position.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.PreventNodeEvent Property

Gets or sets whether to prevent node event from being raised after cell's one.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool PreventNodeEvent { get; set; }
```

Visual Basic Copy Code

```
Public Property PreventNodeEvent As Boolean
```

Property Value

true to stop node event from being raised after cell one, or **false** otherwise. The default value is **false**.

Remarks

This applies to NodeClicked or NodeDoubleClicked events raised for a TableNode, after respective CellClicked or CellDoubleClicked.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.Row Property

Gets the index of the table's row that contains Cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Row { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Row As Integer
```

Property Value

An integer value specifying the zero-based index of the table row related to the event.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellEventArgs.Table Property

Get the table that contains Cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode Table { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Table As TableNode
```

Property Value

An instance of the TableNode class specifying the table related to the event.

See Also

CellEventArgs Members
CellEventArgs Class
MindFusion.Diagramming Namespace

CellFrameStyle Enumeration

Specifies the style of the cells' frame lines.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum CellFrameStyle
```

Visual Basic Copy Code

```
Public Enum CellFrameStyle
```

Members

	Member name	Description
	None	Table cells have no border.
	Simple	The border of table cells is simple line.
	System3D	The border has 3D look.

Remarks

The enumeration values are used when customizing table cells.

See Also

MindFusion.Diagramming Namespace
CellFrameStyle Property

CellValidationEventArgs Class

Contains the arguments passed to event handlers, such as CellTextEditing.

Syntax

C# Copy Code

```
public class CellValidationEventArgs : CellEventArgs
```

Visual Basic — Copy Code

```
Public Class CellValidationEventArgs
    Inherits CellEventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.CellEventArgs
      MindFusion.Diagramming.CellValidationEventArgs

See Also

CellValidationEventArgs Members
MindFusion.Diagramming Namespace

CellValidationEventArgs Members

Contains the arguments passed to event handlers, such as CellTextEditing.

The following tables list the members exposed by the CellValidationEventArgs type.

Public Constructors

	Name	Description
	CellValidationEventArgs	Initializes a new instance of the **CellValidationEventArgs** class.

Public Properties

	Name	Description
	Cancel	Gets or sets a value indicating whether to prevent the user from completing the operation.
	Cell	Get the cell for which the event is raised. (Inherited from CellEventArgs.)
	Column	Gets the index of the table's column that contains **Cell**. (Inherited from CellEventArgs.)
	MouseButton	Gets which mouse button has been pressed. (Inherited from CellEventArgs.)
	MousePosition	Gets the current mouse pointer position. (Inherited from CellEventArgs.)
	PreventNodeEvent	Gets or sets whether to prevent node event from being raised after cell's one. (Inherited from CellEventArgs.)
	Row	Gets the index of the table's row that contains **Cell**. (Inherited from CellEventArgs.)
	Table	Get the table that contains **Cell**. (Inherited from CellEventArgs.)

Public Methods

	Name	Description
	CancelDrag	Cancels the resize of the node's row or column.

See Also

CellValidationEventArgs Class
MindFusion.Diagramming Namespace

CellValidationEventArgs Constructor

Initializes a new instance of the CellValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CellValidationEventArgs (
    TableNode.Cell cell,
    MouseButton mouseButton,
    Point mousePosition,
    int column,
    int row
)
```

Visual Basic Copy Code

```
Public New ( _
    cell As TableNode.Cell, _
    mouseButton As MouseButton, _
    mousePosition As Point, _
    column As Integer, _
    row As Integer _
)
```

Parameters

cell
The TableNode.Cell related to the event.

mouseButton
The mouse button related to the event.

mousePosition
The position of the mouse at the time the event was raised.

column
The zero-based index of the column related to the event.

row
The zero-based index of the row related to the event.

See Also

CellValidationEventArgs Class
MindFusion.Diagramming Namespace

CellValidationEventArgs Methods

CellValidationEventArgs.CancelDrag Method

Cancels the resize of the node's row or column.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void CancelDrag ()
```

Visual Basic Copy Code

```
Public Sub CancelDrag ()
```

Remarks

Calling this method immediately stops the operation.

See Also

CellValidationEventArgs Members
CellValidationEventArgs Class
MindFusion.Diagramming Namespace

CellValidationEventArgs Properties

CellValidationEventArgs.Cancel Property

Gets or sets a value indicating whether to prevent the user from completing the operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Cancel { get; set; }
```

Visual Basic Copy Code

```
Public Property Cancel As Boolean
```

Property Value

true to cancel the operation; **false** to approve it.

See Also

CellValidationEventArgs Members
CellValidationEventArgs Class
MindFusion.Diagramming Namespace

ClipboardEventArgs Class

Contains the arguments passed to the RequestClipboard event handlers.

Syntax

C# Copy Code

```
public class ClipboardEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class ClipboardEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.ClipboardEventArgs

See Also

ClipboardEventArgs Members

Contains the arguments passed to the RequestClipboard event handlers.

The following tables list the members exposed by the ClipboardEventArgs type.

Public Constructors

	Name	Description
	ClipboardEventArgs	Overloaded.

Public Properties

	Name	Description
	Data	Gets or sets the clipboard data.
	DataType	Gets or sets the type of the data to get or set from the clipboard. This should be **null** (**Nothing** in Visual Basic) for clipboard set operations.
	IsGetOperation	Gets a value indicating whether getting or setting the clipboard.
	IsHandled	Gets or sets a value indicating whether the event is handled.
	Source	Gets the object, which initiated the event.

See Also

ClipboardEventArgs Class
MindFusion.Diagramming Namespace

ClipboardEventArgs Constructor

Overload List

Name	Description
ClipboardEventArgs (Object)	Initializes a new instance of the ClipboardEventArgs class for a Clipboard get operation.
ClipboardEventArgs (Object, Object)	Initializes a new instance of the **ClipboardEventArgs** class for a Clipboard set operation.

See Also

ClipboardEventArgs Class
MindFusion.Diagramming Namespace

ClipboardEventArgs Constructor (Object)

Initializes a new instance of the ClipboardEventArgs class for a Clipboard get operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ClipboardEventArgs (
    Object source
)
```

Visual Basic Copy Code

```
Public New ( _
    source As Object _
)
```

Parameters

source
The object requesting the Clipboard contents.

See Also

ClipboardEventArgs Constructor Overload List
ClipboardEventArgs Class
MindFusion.Diagramming Namespace

ClipboardEventArgs Constructor (Object, Object)

Initializes a new instance of the ClipboardEventArgs class for a Clipboard set operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ClipboardEventArgs (
    Object source,
    Object value
)
```

Visual Basic Copy Code

```
Public New ( _
    source As Object, _
    value As Object _
)
```

Parameters

source
The object requesting to set the Clipboard contents.

value
The new contents for the Clipboard.

See Also

ClipboardEventArgs Constructor Overload List
ClipboardEventArgs Class
MindFusion.Diagramming Namespace

ClipboardEventArgs Properties

ClipboardEventArgs.Data Property

Gets or sets the clipboard data.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Data { get; set; }
```

Visual Basic Copy Code

```
Public Property Data As Object
```

Property Value

An object representing the Clipboard contents.

Remarks

The purpose of this property depends on the type of the Clipboard operation, as specified by IsGetOperation. During *get* operations the property will contain the contents of the Clipboard after the event is handled. For *set* operations the property specifies the new contents of the Clipboard.

See Also

ClipboardEventArgs Members
ClipboardEventArgs Class
MindFusion.Diagramming Namespace

ClipboardEventArgs.DataType Property

Gets or sets the type of the data to get or set from the clipboard. This should be **null** (**Nothing** in Visual Basic) for clipboard set operations.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Type DataType { get; set; }
```

Visual Basic Copy Code

```
Public Property DataType As Type
```

Property Value

A **Type** instance.

See Also

ClipboardEventArgs Members
ClipboardEventArgs Class
MindFusion.Diagramming Namespace

ClipboardEventArgs.IsGetOperation Property

Gets a value indicating whether getting or setting the clipboard.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsGetOperation { get; }
```

Visual Basic

```
Public ReadOnly Property IsGetOperation As Boolean
```

Property Value

true if the request is to get the Clipboard contents; **false** if the request is to set Clipboard contents.

Remarks

The value of this property is inferred by the constructor used to create the event argument and cannot be subsequently changed.

See Also

ClipboardEventArgs Members
ClipboardEventArgs Class
MindFusion.Diagramming Namespace

ClipboardEventArgs.IsHandled Property

Gets or sets a value indicating whether the event is handled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool IsHandled { get; set; }
```

Visual Basic

```
Public Property IsHandled As Boolean
```

Property Value

See Also

ClipboardEventArgs Members
ClipboardEventArgs Class
MindFusion.Diagramming Namespace

ClipboardEventArgs.Source Property

Gets the object, which initiated the event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Object Source { get; }
```

Visual Basic

```
Public ReadOnly Property Source As Object
```

Property Value

The object requesting the Clipboard.

See Also

ClipboardEventArgs Members
ClipboardEventArgs Class

CollectionBase(T) Class

An implementation of the **IList<T>** interface.

Syntax

C# Copy Code

```
public abstract class CollectionBase<T> : IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
```

Visual Basic Copy Code

```
Public MustInherit Class CollectionBase(Of T)
    Implements IList(Of T), ICollection(Of T), IEnumerable(Of T), IEnumerable
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.CollectionBase(T)
    MindFusion.Diagramming.AnchorPointCollection
    MindFusion.Diagramming.GroupCollection
    MindFusion.Diagramming.ItemCollectionBase(T)

See Also

CollectionBase(T) Members
MindFusion.Diagramming Namespace

CollectionBase(T) Members

An implementation of the **IList<T>** interface.

The following tables list the members exposed by the CollectionBase(T) type.

Protected Constructors

	Name	Description
	CollectionBase	Initializes a new instance of the **CollectionBase<T>** class.

Protected Fields

	Name	Description
	InnerList	Gets a **List<T>** containing the list of elements in the collection instance.

Public Properties

	Name	Description
	Count	Gets the number of elements actually contained in the collection.
	IsReadOnly	Gets a value indicating whether the collection is read-only.
	Item	Gets or sets the element at the specified index.

Public Methods

	Name	Description
	Add	Adds an object to the end of the collection.

	Clear	Removes all elements from the collection.
	Contains	Determines whether an element is in the collection.
	CopyTo	Copies the entire collection to a compatible one-dimensional array, starting at the specified index of the target array.
	GetEnumerator	Returns an enumerator that iterates through the collection.
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire collection.
	Insert	Inserts an element into the collection at the specified index.
	Remove	Removes the first occurrence of a specific object from the collection.
	RemoveAt	Removes the element at the specified index of the collection.
	Sort	Overloaded.

See Also

CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T) Fields

CollectionBase(T).InnerList Field

Gets a **List<T>** containing the list of elements in the collection instance.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected List<T> InnerList
```

Visual Basic Copy Code

```
Protected InnerList As List(Of T)
```

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T) Constructor

Initializes a new instance of the CollectionBase<T> class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected CollectionBase<T> ()
```

Visual Basic Copy Code

```
Protected Sub New(Of T) ()
```

See Also

CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T) Methods

CollectionBase(T).Add Method

Adds an object to the end of the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract void Add (
    T item
)
```

Visual Basic Copy Code

```
Public MustOverride Sub Add( _
    item As T _
)
```

Parameters

item
The object to be added to the end of the collection.

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T).Clear Method

Removes all elements from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract void Clear ()
```

Visual Basic Copy Code

```
Public MustOverride Sub Clear ()
```

See Also

CollectionBase(T) Members
CollectionBase(T) Class

CollectionBase(T).Contains Method

Determines whether an element is in the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Contains (
    T item
)
```

Visual Basic Copy Code

```
Public Function Contains( _
    item As T _
) As Boolean
```

Parameters

item
The object to locate in the collection.

Return Value

true if item is found in the collection; otherwise, **false**.

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T).CopyTo Method

Copies the entire collection to a compatible one-dimensional array, starting at the specified index of the target array.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void CopyTo (
    T[] array,
    int arrayIndex
)
```

Visual Basic Copy Code

```
Public Sub CopyTo( _
    array() As T, _
    arrayIndex As Integer _
)
```

Parameters

array
The one-dimensional array that is the destination of the elements copied from collection.

arrayIndex
The zero-based index in array at which copying begins.

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T).GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IEnumerator<T> GetEnumerator ()
```

Visual Basic Copy Code

```
Public Function GetEnumerator () As IEnumerator(Of T)
```

Return Value

An **IEnumerator<T>** for the collection.

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T).IndexOf Method

Searches for the specified object and returns the zero-based index of the first occurrence within the entire collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int IndexOf (
    T item
)
```

Visual Basic Copy Code

```
Public Function IndexOf( _
    item As T _
) As Integer
```

Parameters

item
The object to locate in the collection.

Return Value

The zero-based index of the first occurrence of item within the entire collection, if found; otherwise, –1.

See Also

CollectionBase(T).Insert Method

Inserts an element into the collection at the specified index.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract void Insert (
    int index,
    T item
)
```

Visual Basic Copy Code

```
Public MustOverride Sub Insert( _
    index As Integer, _
    item As T _
)
```

Parameters

index
The zero-based index at which item should be inserted.

item
The object to insert.

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T).Remove Method

Removes the first occurrence of a specific object from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract bool Remove (
    T item
)
```

Visual Basic Copy Code

```
Public MustOverride Function Remove( _
    item As T _
) As Boolean
```

Parameters

item
The object to remove from the collection.

Return Value

true if item is successfully removed; otherwise, **false**.

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T).RemoveAt Method

Removes the element at the specified index of the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract void RemoveAt (
    int index
)
```

Visual Basic Copy Code

```
Public MustOverride Sub RemoveAt( _
    index As Integer _
)
```

Parameters

index
The zero-based index of the element to remove.

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T).Sort Method

Overload List

Name	Description
CollectionBase(T).Sort (Comparison(T))	Sorts the elements in this collection using the specified comparison.
CollectionBase(T).Sort (IComparer(T))	Sorts the elements in this collection using the specified comparer.

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T).Sort Method (Comparison(T))

Sorts the elements in this collection using the specified comparison.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Sort (
    Comparison<T> comparison
)
```

Visual Basic Copy Code

```
Public Sub Sort( _
    comparison As Comparison(Of T) _
)
```

Parameters

comparison
The **System.Comparison** to use when comparing elements.

See Also

Sort Method Overload List
CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T).Sort Method (IComparer(T))

Sorts the elements in this collection using the specified comparer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Sort (
    IComparer<T> comparer
)
```

Visual Basic Copy Code

```
Public Sub Sort( _
    comparer As IComparer(Of T) _
)
```

Parameters

comparer
The **IComparer** implementation to use when comparing elements.

See Also

Sort Method Overload List
CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T) Properties

CollectionBase(T).Count Property

Gets the number of elements actually contained in the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Count { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Count As Integer
```

Property Value

An integer value.

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T).IsReadOnly Property

Gets a value indicating whether the collection is read-only.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsReadOnly { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property IsReadOnly As Boolean
```

Property Value

true if the collection is read-only; otherwise, **false**.

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

CollectionBase(T).Item Property

Gets or sets the element at the specified index.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public T Item [
    int index
] { get; set; }
```

Visual Basic Copy Code

```
Public Property Item( _
    index As Integer _
) As T
```

Parameters

index
The zero-based index of the element to get or set.

Property Value

The element at the specified index.

See Also

CollectionBase(T) Members
CollectionBase(T) Class
MindFusion.Diagramming Namespace

ColumnStyle Enumeration

Specifies how table columns' width is set.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum ColumnStyle
```

Visual Basic Copy Code

```
Public Enum ColumnStyle
```

Members

	Member name	Description
	FixedWidth	The columns has fixed, unchangeable width.
	AutoWidth	The columns' width changes when the table is resized.

Remarks

Use the enumeration values to set the behavior of table columns, when the user resizes the table.

See Also

MindFusion.Diagramming Namespace
ColumnStyle Property

CommandHistory Class

Maintains a queue of Command records used to undo or redo actions.

Syntax

C# Copy Code

```
public class CommandHistory
```

Visual Basic Copy Code

```
Public Class CommandHistory
```

Remarks

The only instance of **CommandHistory** can be accessed through the History property of UndoManager.

The Capacity property specifies how many Command instances can be stored in the history queue. If the queue is full and a new action is recorded, the oldest one will be removed and made unavailable for undo/redo. A read-only access to the command queue is provided by the Commands property.

The **CommandHistory** instance maintains an internal record pointer. The pointer is decreased or increased by one when an action is respectively undone or redone through the undo manager. The pointer actually refers to two adjacent action records - the older one to be used when undoing, the newer one when redoing commands. Those two records can be accessed by NextUndo and NextRedo properties. When no action has been undone, the pointer refers to the end of the queue and **NextRedo** returns **null** (**Nothing** in Visual Basic). When all actions have been undone, the pointer refers to the start of the queue and **NextUndo** returns **null**.

Call Clear to remove all action records that are currently in the history queue, making those actions unavailable for undo or redo anymore. That won't disable adding new records to the queue, so it will be possible to undo/redo any actions executed after **Clear**.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.CommandHistory

See Also

CommandHistory Members
MindFusion.Diagramming Namespace

CommandHistory Members

Maintains a queue of Command records used to undo or redo actions.

The following tables list the members exposed by the CommandHistory type.

Public Properties

	Name	Description
	Capacity	Gets or sets the maximum number of records to keep in the history queue.
	Commands	Gets the commands saved in the undo / redo history queue.
	NextRedo	Gets a reference to the command that would be redone if Redo is invoked.
	NextUndo	Gets a reference to the command that would be undone if Undo is invoked.

Public Methods

	Name	Description
	Clear	Removes all records from the history queue.
	MergeUndoRecords	Merges several undo records into a single composite record.

See Also

CommandHistory Class
MindFusion.Diagramming Namespace

CommandHistory Methods

CommandHistory.Clear Method

Removes all records from the history queue.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Clear ()
```

Visual Basic Copy Code

```
Public Sub Clear ()
```

Remarks

Call **Clear** to remove all action records that are currently in the history queue, making those actions unavailable for undo or redo anymore. That won't disable adding new records to the queue, so it will be possible to undo/redo any actions executed after the call.

See Also

CommandHistory Members
CommandHistory Class
MindFusion.Diagramming Namespace
UndoEnabled Property

CommandHistory.MergeUndoRecords Method

Merges several undo records into a single composite record.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool MergeUndoRecords (
    int numRecords,
    string title
)
```

Visual Basic Copy Code

```
Public Function MergeUndoRecords( _
    numRecords As Integer, _
    title As String _
) As Boolean
```

Parameters

numRecords

Specifies how many records at the end of the command list should be merged.

title

Specifies the title of the new command.

Return Value

false if the commands saved in the history are less than *numRecords*; otherwise, **true**.

See Also

CommandHistory Members
CommandHistory Class

CommandHistory Properties

CommandHistory.Capacity Property

Gets or sets the maximum number of records to keep in the history queue.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Capacity { get; set; }
```

Visual Basic Copy Code

```
Public Property Capacity As Integer
```

Property Value

An integer value specifying the number of records to keep in the queue; cannot be less than 1. The default is 10.

Remarks

If the history queue is full, the oldest command records are removed from the beginning when new ones are added. The queue is cleared when you change **Capacity**, so better do this at application's startup phase.

See Also

CommandHistory Members
CommandHistory Class
MindFusion.Diagramming Namespace

CommandHistory.Commands Property

Gets the commands saved in the undo / redo history queue.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CommandList Commands { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Commands As CommandList
```

Property Value

An instance of the CommandList class representing a read-only collection of Command objects.

See Also

CommandHistory Members
CommandHistory Class
MindFusion.Diagramming Namespace
NextUndo Property
NextRedo Property

CommandHistory.NextRedo Property

Gets a reference to the command that would be redone if Redo is invoked.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Command NextRedo { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NextRedo As Command
```

Property Value

An instance of the Command class.

Remarks

When no action has been undone, the current action pointer refers to the end of the history queue and **NextRedo** returns **null** (**Nothing** in Visual Basic).

See Also

CommandHistory Members
CommandHistory Class
MindFusion.Diagramming Namespace
Redo Method
NextUndo Property

CommandHistory.NextUndo Property

Gets a reference to the command that would be undone if Undo is invoked.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Command NextUndo { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NextUndo As Command
```

Property Value

An instance of the Command class.

Remarks

When all actions have been undone, the current action pointer refers to the start of the history queue and **NextUndo** returns **null** (**Nothing** in Visual Basic).

See Also

CommandHistory Members
CommandHistory Class
MindFusion.Diagramming Namespace
Undo Method
NextRedo Property

CompositeNode Class

CompositeNode implements nodes whose appearance can be defined via composition of components and layout containers.

Syntax

C# Copy Code

```
public class CompositeNode : DiagramNode, IComponentHost
```

Visual Basic Copy Code

```
Public Class CompositeNode
    Inherits DiagramNode
    Implements IComponentHost
```

Remarks

Use the Components property to add components to the node. The components are arranged on top of each other by default. To apply custom arrangement logic either parent the components in a ContainerComponent, such as GridPanel, and add this **ContainerComponent** to the **CompositeNode** or override the ArrangeComponents method to apply custom arrangement logic.

To search for a component given its name, use the FindComponent method. To get a component at a specific position, use GetComponentAt.

For additional information about components, check the **Components** topic.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.CompositeNode

See Also

CompositeNode Members
MindFusion.Diagramming Namespace

CompositeNode Members

CompositeNode implements nodes whose appearance can be defined via composition of components and layout containers.

The following tables list the members exposed by the **CompositeNode** type.

Public Constructors

	Name	Description
	CompositeNode	Overloaded. Initializes a new instance of the **CompositeNode** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)

	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	ClipToOutline	Gets or sets a value specifying whether contents will clip to outline, when an outline component is used.
	Components	Gets a list with the root components contained in the node.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)

	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)

	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	ArrangeComponents	Invokes when the components within this node are being arranged.
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsComponent	Checks whether the specified component is contained in this node.
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DocToComponent	Converts a point from document coordinates to the local coordinate space of a component inside this node.
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Draws the node on the specified IGraphics surface.
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified **IGraphics** surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Draws the shadow of this node on the specified IGraphics surface.

	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	FindComponent	Returns the child component with the specified name or **null** (**Nothing** in Visual Basic), if the component was not found.
	GenerateJsonTemplate	Generates a Json template from current component hierarchy.
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetComponentAt	Returns the component at the specified point, expressed in document coordinates.
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	InvalidateLayout	Invalidates the component layout, causing rearrangement of content.
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	LoadJsonTemplate	Loads the item's components from a Json template.
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)

	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	PointToComponent	For internal use.
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from

		DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a IGraphicsPath object for this node as though it is contained in the specified rectangle.
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object.
	LoadFromXml	DiagramItem.LoadFromXml override.
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override.
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramNode.OnRemove override.
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	DiagramNode.OnUpdateBounds override.
	OnZoomChanged	DiagramItem.OnZoomChanged override.

	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object.
	SaveToXml	DiagramItem.SaveToXml override.
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SerializeComponents	Invoked to query whether the underlying component structure should be automatically serialized.
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode Constructor

Initializes a new instance of the CompositeNode class.

Overload List

Name	Description
CompositeNode ()	Initializes a new instance of the **CompositeNode** class.
CompositeNode (CompositeNode)	Initializes a new instance of the **CompositeNode** class using the specified diagram item as a prototype.
CompositeNode (Diagram)	Initializes a new instance of the **CompositeNode** class with default values supplied form the specified Diagram.

See Also

CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode Constructor ()

Initializes a new instance of the CompositeNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CompositeNode ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

CompositeNode Constructor Overload List
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode Constructor (CompositeNode)

Initializes a new instance of the CompositeNode class using the specified diagram item as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CompositeNode (
    CompositeNode prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As CompositeNode _
)
```

Parameters

prototype
The item to use as a prototype. The properties of the new item are initialized from the property values of this item.

Remarks

This constructor copies the DiagramNode property values from the specified prototype node.

See Also

CompositeNode Constructor Overload List
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode Constructor (Diagram)

Initializes a new instance of the CompositeNode class with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CompositeNode (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram _
)
```

Parameters

diagram
The Diagram from which to obtain default values for the node properties.

Remarks

This constructor initializes the CompositeNode properties with the values specified in the respective default-value properties of the Diagram object.

See Also

CompositeNode Constructor Overload List
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode Methods

CompositeNode.ArrangeComponents Method

Invokes when the components within this node are being arranged.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void ArrangeComponents (
    Rect availableSpace,
    IGraphics graphics
)
```

Visual Basic Copy Code

```
Public Overridable Sub ArrangeComponents( _
    availableSpace As Rect, _
    graphics As IGraphics _
)
```

Parameters

availableSpace
The available space.

graphics
An IGraphics object that can be used for text measurement.

Remarks

By default components are arranged on top of each other, in the order they were added to the Components collection. To apply more specialized arrangement add the components to a ContainerComponent (such as SimplePanel, StackPanel or GridPanel) then add this **ContainerComponent** as a child of the CompositeNode. The component system allows you to implement additional **ContainerComponent**-derived classes to apply custom arrangement logic.

Another way to apply custom arrangement is to derive from **CompositeNode** and override the ArrangeComponents method. Then position the components by setting their Bounds property.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.ContainsComponent Method

Checks whether the specified component is contained in this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ContainsComponent (
    ComponentBase component
)
```

Visual Basic Copy Code

```
Public Function ContainsComponent( _
    component As ComponentBase _
) As Boolean
```

Parameters

component
The component to check.

Return Value

true if the component is contained in the node; otherwise, false.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.DocToComponent Method

Converts a point from document coordinates to the local coordinate space of a component inside this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point DocToComponent (
    Point docPoint,
    ComponentBase component
)
```

Visual Basic Copy Code

```
Public Function DocToComponent( _
    docPoint As Point, _
    component As ComponentBase _
) As Point
```

Parameters

docPoint
The point to convert.

component
The target component.

Return Value

A **Microsoft.Maui.Graphics.Point** instance representing the point in local coordinates.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.DrawLocal Method

Draws the node on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawLocal (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawLocal( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
The IGraphics surface to draw the node on.

options
Provides options for the drawing operation.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.DrawShadowLocal Method

Draws the shadow of this node on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawShadowLocal (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawShadowLocal( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
An IGraphics object where the shadow should be rendered.

options
Provides options for the drawing operation.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.FindComponent Method

Returns the child component with the specified name or **null** (**Nothing** in Visual Basic), if the component was not found.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ComponentBase FindComponent (
    string name
)
```

Visual Basic Copy Code

```
Public Function FindComponent( _
    name As String _
) As ComponentBase
```

Parameters

name

The name of the component to search for.

Return Value

An instance of the ComponentBase class identifying the found component or **null** (**Nothing** in Visual Basic).

Remarks

This method is useful for obtaining a references to components defined in XML.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.GenerateJsonTemplate Method

Generates a Json template from current component hierarchy.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string GenerateJsonTemplate ()
```

Visual Basic Copy Code

```
Public Function GenerateJsonTemplate () As String
```

Return Value

A Json template string containing the item's components data.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.GetComponentAt Method

Returns the component at the specified point, expressed in document coordinates.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual ComponentBase GetComponentAt (
    Point point
)
```

Visual Basic Copy Code

```
Public Overridable Function GetComponentAt( _
    point As Point _
) As ComponentBase
```

Parameters

point

A **Microsoft.Maui.Graphics.Point** specifying the point to hit-test, expressed in document coordinates.

Return Value

An instance of the ComponentBase class identifying the topmost component at the specified point, or **null** if no component found.

Remarks

This method ignores components with any of the following properties set to **false**: IsEnabled, IsHitTestVisible, IsInteractive.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.GetRgnInRect Method

Creates and returns a IGraphicsPath object for this node as though it is contained in the specified rectangle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override IGraphicsPath GetRgnInRect (
    Rect rc,
    bool inLocal
)
```

Visual Basic Copy Code

```
Protected Overrides Function GetRgnInRect( _
    rc As Rect, _
    inLocal As Boolean _
) As IGraphicsPath
```

Parameters

rc

inLocal

Return Value

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.InvalidateLayout Method

Invalidates the component layout, causing rearrangement of content.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void InvalidateLayout ()
```

Visual Basic Copy Code

```
Public Sub InvalidateLayout ()
```

Remarks

Calling this method will cause ArrangeComponents to be invoked before the next drawing pass.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.LoadFromJson Method

Loads the node from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the node's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

Remarks

By default, all components of a CompositeNode are saved in the JSON document along with their associated data. This will lead to wasted space if the instances of a derived class are built upon the same hierarchy of components and expose only a few properties corresponding to component attributes. The recommended approach of serializing such classes is to override SerializeComponents and return **false** to stop the built-in serialization of components, and save / load the node properties from SaveToJson / **LoadFromJson** overrides.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.LoadFromXml Method

DiagramItem.LoadFromXml override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
```

```
    XmlPersistContext context
)
```

Visual Basic — Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** containing the item's data.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

Remarks

By default, all components of a CompositeNode are saved in the XML document along with their associated data. This will lead to wasted space if the instances of a derived class are built upon the same hierarchy of components and expose only a few properties corresponding to component attributes. The recommended approach of serializing such classes is to override SerializeComponents and return **false** to stop the built-in serialization of components, and save / load the node properties from SaveToXml / **LoadFromXml** overrides. This is demonstrated in the *Scripting* sample project.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.LoadJsonTemplate Method

Loads the item's components from a Json template.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# — Copy Code

```
public void LoadJsonTemplate (
    string template
)
```

Visual Basic — Copy Code

```
Public Sub LoadJsonTemplate( _
    template As String _
)
```

Parameters

template

A Json template string containing the item's components data.

Example

Components can be defined in JSON format as shown in this example:

C# — Copy Code

```
string template = @"{
""component"": ""SimplePanel"",
""children"":
[
 {
  ""component"": ""Shape"",
```

```
  ""name"": ""Frame"",
  ""pen"": ""black"",
  ""strokeThickness"": 0.2,
  ""isOutline"": true,
  ""autoProperty"": true
 },
{
""component"": ""GridPanel"",
""rowDefinitions"": [""*""],
""columnDefinitions"": [""22"", ""*""],
""children"":
[
 {
  ""component"": ""Image"",
  ""name"": ""Image"",
  ""autoProperty"": true,
  ""location"": ""../Images/icon4.png"",
  ""margin"": ""1"",
  ""imageAlign"": ""Fit""
 },
       {
         ""component"": ""Button"",
        ""brush"": ""red"",
        ""textColor"": ""#fff"",
        ""font"": ""Arial bold"",
        ""text"": ""x"",
        ""cornerRadius"": 3.5,
        ""width"": 6,
        ""height"": 6,
        ""clickHandler"": ""onDeleteClick""
       },
 {
  ""component"": ""StackPanel"",
  ""orientation"": ""Vertical"",
  ""gridColumn"": 1,
  ""margin"": ""1"",
  ""verticalAlignment"": ""Near"",
  ""children"":
  [
   {
    ""component"": ""Text"",
    ""name"": ""Title"",
    ""autoProperty"": true,
    ""text"": ""title"",
    ""font"": ""Arial bold""
   },
   {
    ""component"": ""Text"",
    ""name"": ""FullName"",
    ""autoProperty"": true,
    ""text"": ""full name"",
    ""pen"": ""blue"",
    ""padding"": ""1,0,1,0"",
    ""enableStyledText"": true
   },
   {
    ""component"": ""Text"",
    ""name"": ""Details"",
    ""autoProperty"": true,
    ""text"": ""details"",
    ""font"": ""Arial 3""
   }
  ]
 }
]
}]}
"
```

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.OnMeasureUnitChanged Method

DiagramItem.OnMeasureUnitChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnMeasureUnitChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnMeasureUnitChanged ()
```

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.OnRemove Method

DiagramNode.OnRemove override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnRemove ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnRemove ()
```

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.OnUpdateBounds Method

DiagramNode.OnUpdateBounds override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnUpdateBounds (
    bool sizeChanged
)
```

Visual Basic Copy Code

```
Protected Overrides Sub OnUpdateBounds( _
    sizeChanged As Boolean _
)
```

Parameters

sizeChanged
true if the size has changed; **false** if only the position has changed.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.OnZoomChanged Method

DiagramItem.OnZoomChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnZoomChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnZoomChanged ()
```

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.PointToComponent Method

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point PointToComponent (
    Point point
)
```

Visual Basic Copy Code

```
Public Function PointToComponent( _
    point As Point _
) As Point
```

Parameters

point

Return Value

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.SaveToJson Method

Saves the node as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject instance to contain the serialized data.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

Remarks

By default, all components of a CompositeNode are saved in the JSON document along with their associated data. This will lead to wasted space if the instances of a derived class are built upon the same hierarchy of components and expose only a few properties corresponding to component attributes. The recommended approach of serializing such classes is to override SerializeComponents and return **false** to stop the built-in serialization of components, and save / load the node properties from **SaveToJson** / LoadFromJson overrides.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.SaveToXml Method

DiagramItem.SaveToXml override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** the item's data should be stored into.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

Remarks

By default, all components of a CompositeNode are saved in the XML document along with their associated data. This will lead to wasted space if the instances of a derived class are built upon the same hierarchy of components and expose only a few properties corresponding to component attributes. The recommended approach of serializing such classes is to override SerializeComponents and return **false** to stop the built-in serialization of components, and save / load the node properties from **SaveToXml** / LoadFromXml overrides. This is demonstrated in the *Scripting* sample project.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.SerializeComponents Method

Invoked to query whether the underlying component structure should be automatically serialized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual bool SerializeComponents ()
```

Visual Basic Copy Code

```
Protected Overridable Function SerializeComponents () As Boolean
```

Return Value

true to automatically serialize the component tree; **false** to skip the automatic component serialization.

Remarks

By default, all components of a CompositeNode are saved along with their associated data. This will lead to wasted space if the instances of a derived class are built upon the same hierarchy of components and expose only a few properties corresponding to component attributes. The recommended approach of serializing such classes is to override **SerializeComponents** and return **false** to stop the built-in serialization of components. Then you can save or load only the necessary node properties from SaveTo / LoadFrom overrides for binary format, or SaveToXml / LoadFromXml for XML format. This is demonstrated in *EmployeeNode.cs* in the *Demo* sample project.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode Properties

CompositeNode.ClipToOutline Property

Gets or sets a value specifying whether contents will clip to outline, when an outline component is used.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ClipToOutline { get; set; }
```

Visual Basic Copy Code

```
Public Property ClipToOutline As Boolean
```

Property Value

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNode.Components Property

Gets a list with the root components contained in the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IList<ComponentBase> Components { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Components As IList(Of ComponentBase)
```

Property Value

A list with the components contained in this node.

See Also

CompositeNode Members
CompositeNode Class
MindFusion.Diagramming Namespace

CompositeNodeStyle Class

Represents a reusable set of properties that define the appearance of composite nodes.

Syntax

C# Copy Code

```
public class CompositeNodeStyle : DiagramNodeStyle
```

Visual Basic — Copy Code

```
Public Class CompositeNodeStyle
    Inherits DiagramNodeStyle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle
      MindFusion.Diagramming.DiagramNodeStyle
        MindFusion.Diagramming.CompositeNodeStyle

See Also

CompositeNodeStyle Members
MindFusion.Diagramming Namespace

CompositeNodeStyle Members

Represents a reusable set of properties that define the appearance of composite nodes.

The following tables list the members exposed by the CompositeNodeStyle type.

Public Constructors

	Name	Description
	CompositeNodeStyle	Initializes a new instance of the **CompositeNodeStyle** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of the item. (Inherited from DiagramItemStyle.)
	Effects	Gets a list of the effects applied to the node. (Inherited from DiagramNodeStyle.)
	FontFamily	Gets or sets the font family of the item text. (Inherited from DiagramItemStyle.)
	FontSize	Gets or sets the font size of the item text. (Inherited from DiagramItemStyle.)
	FontStyle	Gets or sets the font style of the item text. (Inherited from DiagramItemStyle.)
	FontUnit	Gets or sets the font unit of the item text. (Inherited from DiagramItemStyle.)
	ShadowBrush	Gets or sets the brush used to paint the shadow of the item. (Inherited from DiagramItemStyle.)
	Stroke	Gets or sets the brush used to paint the outline of the item. (Inherited from DiagramItemStyle.)
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items. (Inherited from DiagramItemStyle.)
	StrokeThickness	Gets or sets the thickness of the item's outline, expressed in the current units of measure. (Inherited from DiagramItemStyle.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItemStyle.)

Public Methods

	Name	Description
	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
S	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JsonObject. (Inherited from DiagramNodeStyle.)
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)
	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a JsonObject. (Inherited from DiagramNodeStyle.)
	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

CompositeNodeStyle Class
MindFusion.Diagramming Namespace

CompositeNodeStyle Constructor

Initializes a new instance of the CompositeNodeStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CompositeNodeStyle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

CompositeNodeStyle Events

Style.PropertyChanged Event

Raised when any of the registered style properties has changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<StylePropertyChangedEventArgs> PropertyChanged
```

Visual Basic Copy Code

```
Public Event PropertyChanged As EventHandler(Of StylePropertyChangedEventArgs)
```

Event Data

PropertyChanged event handlers receive an argument of type StylePropertyChangedEventArgs.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

CompositeRouter Class

Implements a chain of link routers (objects implementing ILinkRouter interface) so that a link passes through the sequence until it gets routed successfully.

Syntax

C# Copy Code

```
public class CompositeRouter : ILinkRouter
```

Visual Basic Copy Code

```
Public Class CompositeRouter
    Implements ILinkRouter
```

Remarks

By default diagram's LinkRouter is set to a **CompositeRouter** instance that chains PatternRouter and GridRouter. You can insert additional link routers into the Routers list, including custom implementations of ILinkRouter if target diagram should favor some specific link paths.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.CompositeRouter

See Also

CompositeRouter Members
MindFusion.Diagramming Namespace

CompositeRouter Members

Implements a chain of link routers (objects implementing ILinkRouter interface) so that a link passes through the sequence until it gets routed successfully.

The following tables list the members exposed by the CompositeRouter type.

Public Constructors

	Name	Description
	CompositeRouter	Overloaded.

Public Properties

	Name	Description
	Diagram	Gets or sets the Diagram whose links will be routed.
	Routers	A list of router objects forming a chain through which links pass until they are routed successfully.

Public Methods

	Name	Description
	RouteLink	Implements ILinkRouter.RouteLink.
	RouteLinks	Implements ILinkRouter.RouteLinks.

See Also

CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter Constructor

Overload List

Name	Description
CompositeRouter ()	Initializes a new instance of the CompositeRouter class.
CompositeRouter (Diagram)	Initializes a new instance of the CompositeRouter class.

See Also

CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter Constructor ()

Initializes a new instance of the CompositeRouter class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CompositeRouter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

CompositeRouter Constructor Overload List
CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter Constructor (Diagram)

Initializes a new instance of the CompositeRouter class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CompositeRouter (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram _
)
```

Parameters

diagram
The Diagram whose links will be routed.

See Also

CompositeRouter Constructor Overload List
CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter Methods

CompositeRouter.RouteLink Method

Implements ILinkRouter.RouteLink.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RouteLink (
    DiagramLink link,
    ConnectionPoint originConnection,
    ConnectionPoint destinationConnection,
    bool nowCreating
)
```

Visual Basic Copy Code

```
Public Function RouteLink( _
    link As DiagramLink, _
    originConnection As ConnectionPoint, _
    destinationConnection As ConnectionPoint, _
    nowCreating As Boolean _
) As Boolean
```

Parameters

link

The DiagramLink to route.

originConnection

Represents connection to Origin node, potentially different from link's stored connection when modifying first control point.

destinationConnection

Represents connection to Destination node, potentially different from link's stored connection when modifying last control point.

nowCreating

Indicates whether link is currently being created.

Return Value

true if a path has been found successfully, otherwise false.

See Also

CompositeRouter Members
CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter.RouteLinks Method

Implements ILinkRouter.RouteLinks.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection RouteLinks (
    DiagramLinkCollection links
)
```

Visual Basic Copy Code

```
Public Function RouteLinks( _
    links As DiagramLinkCollection _
) As DiagramLinkCollection
```

Parameters

links

A list of DiagramLink objects that should be routed.

Return Value

A list of links that were not routed successfully.

See Also

CompositeRouter Members
CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter Properties

CompositeRouter.Diagram Property

Gets or sets the Diagram whose links will be routed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Diagram Diagram { get; set; }
```

Visual Basic Copy Code

```
Public Property Diagram As Diagram
```

Property Value

An instance of the Diagram class.

See Also

CompositeRouter Members
CompositeRouter Class
MindFusion.Diagramming Namespace

CompositeRouter.Routers Property

A list of router objects forming a chain through which links pass until they are routed successfully.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<ILinkRouter> Routers { get; set; }
```

Visual Basic Copy Code

```
Public Property Routers As List(Of ILinkRouter)
```

Property Value

A list of objects implementing ILinkRouter interface.

Remarks

By default the list contains PatternRouter and GridRouter objects.

See Also

CompositeRouter Members
CompositeRouter Class
MindFusion.Diagramming Namespace

ConnectionPoint Class

This class represents a connection between an edge and a node in a graph. It decouples the DiagramLink class from the DiagramNode class, allowing edges to connect different types of nodes without knowing anything of their implementation details. Each **DiagramLink** object contains two **ConnectionPoint** instances describing the connections at both ends of the link.

Syntax

C# Copy Code

```
public class ConnectionPoint
```

Visual Basic Copy Code

```
Public Class ConnectionPoint
```

Remarks

If custom nodes should store additional information about their connections, inherit the **ConnectionPoint** class, add the necessary members and return an instance of this class from the CreateConnectionPoint override. For example, TreeViewNode uses TreeViewConnectionPoint objects to store reference to the item associated with the connection.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ConnectionPoint
    MindFusion.Diagramming.TreeViewConnectionPoint
    MindFusion.Diagramming.TableConnectionPoint
    MindFusion.Diagramming.TreeViewConnectionPoint

See Also

ConnectionPoint Members
MindFusion.Diagramming Namespace

ConnectionPoint Members

This class represents a connection between an edge and a node in a graph. It decouples the DiagramLink class from the DiagramNode class, allowing edges to connect different types of nodes without knowing anything of their implementation details. Each **DiagramLink** object contains two ConnectionPoint instances describing the connections at both ends of the link.

The following tables list the members exposed by the **ConnectionPoint** type.

Public Constructors

	Name	Description
	ConnectionPoint	Overloaded. Initializes a new instance of the **ConnectionPoint** class.

Public Properties

	Name	Description
	RelativePosition	Gets or sets the position of this connection point, relative to the associated node's bounding rectangle, expressed as percentage.
	Row	Gets the index of a constituent item within the associated node where the link represented by this connection point is connected, or -1, if the link is connected to the node itself.

Protected Properties

	Name	Description
	Incoming	Gets a flag indicating whether this connection point is at the origin or the destination of the associated link.
	Link	Gets the link associated with this connection point.
	Node	Gets the node associated with this connection point.

Public Methods

	Name	Description

	AddLinkToNode	Adds the link associated with this connection point to the associated node.
	CalculateIntersections	Determines whether the end of a link should be aligned to the point where the node contour is intersected by the line connecting the centers of the connected nodes.
	CanMoveLink	Checks whether the specified point, expressed in document coordinates, is a valid location for this connection point.
	GetAnchorPos	Overloaded.
	GetEndPoint	Calculates the coordinates of the this connection point expressed in document coordinates.
	GetInitialPoint	Returns a point (usually the center of the bounding rectangle) contained within the associated node.
	GetIntersection	Calculates the intersection point between the specified line segment and the contour of the node associated with this connection point.
	GetNodeRect	Returns the bounding rectangle of the node associated with this connection point.
	LinkChanges	Checks whether the relationship represented by the associated link would change if the link is attached to the specified node at the specified point.
	NearestAnchorPoint	Returns the anchor point closest to the specified point from the node's associated anchor pattern.
	NodesIntersect	Checks whether the node associated with the current connection point and the node associated with the specified connection point intersect.
	RemoveLinkFromNode	Removes the link associated with this connection point from the associated node.
	SameNode	Checks whether the current connection point and the specified connection point are associated with the same node.
	SaveEndRelative	Updates the coordinates of the connection point represented by this link, relative to the bounding rectangle of the associated node.

Protected Methods

	Name	Description
	GetNearestBorderPoint	Returns the nearest point lying on the node border.
	UpdateFromPoint	Updates the connection state according to the specified point, without changing the related node.

See Also

ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint Constructor

Initializes a new instance of the ConnectionPoint class.

Overload List

Name	Description
ConnectionPoint ()	Initializes a new instance of the **ConnectionPoint** class.
ConnectionPoint (DiagramNode, DiagramLink, Boolean)	Initializes a new instance of the **ConnectionPoint** class between the specified node and link.

See Also

ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint Constructor ()

Initializes a new instance of the ConnectionPoint class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected ConnectionPoint ()
```

Visual Basic Copy Code

```
Protected Sub New ()
```

Remarks

This constructor is used only during deserialization.

See Also

ConnectionPoint Constructor Overload List
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint Constructor (DiagramNode, DiagramLink, Boolean)

Initializes a new instance of the ConnectionPoint class between the specified node and link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ConnectionPoint (
    DiagramNode node,
    DiagramLink link,
    bool incoming
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode, _
    link As DiagramLink, _
    incoming As Boolean _
)
```

Parameters

node
The node whose connection to a link is managed by this ConnectionPoint.

link
The link whose connection to a node is managed by this **ConnectionPoint**.

incoming

Specifies which end of the link is represented by this **ConnectionPoint**.

See Also

ConnectionPoint Constructor Overload List
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint Methods

ConnectionPoint.AddLinkToNode Method

Adds the link associated with this connection point to the associated node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void AddLinkToNode ()
```

Visual Basic Copy Code

```
Public Overridable Sub AddLinkToNode ()
```

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.CalculateIntersections Method

Determines whether the end of a link should be aligned to the point where the node contour is intersected by the line connecting the centers of the connected nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool CalculateIntersections ()
```

Visual Basic Copy Code

```
Public Overridable Function CalculateIntersections () As Boolean
```

Return Value

true to align the link end; otherwise, **false**.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.CanMoveLink Method

Checks whether the specified point, expressed in document coordinates, is a valid location for this connection point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool CanMoveLink (
    Point point
)
```

Visual Basic Copy Code

```
Public Overridable Function CanMoveLink( _
    point As Point _
) As Boolean
```

Parameters

point
The Point to check for validity.

Return Value

true if the specified point represents a valid location for the connection point; otherwise, **false**.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.GetAnchorPos Method

Overload List

Name	Description
ConnectionPoint.GetAnchorPos (Int32)	Returns the point from the node's associated anchor pattern (if any) with the specified index or the link's initial point.
ConnectionPoint.GetAnchorPos (Int32, Point@)	Retrieves the point from the node's associated anchor pattern (if any) with the specified index.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.GetAnchorPos Method (Int32)

Returns the point from the node's associated anchor pattern (if any) with the specified index or the link's initial point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Point GetAnchorPos (
    int anchorIdx
)
```

Visual Basic Copy Code

```
Public Overridable Function GetAnchorPos( _
    anchorIdx As Integer _
) As Point
```

Parameters

anchorIdx

Return Value

See Also

GetAnchorPos Method Overload List
ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.GetAnchorPos Method (Int32, Point@)

Retrieves the point from the node's associated anchor pattern (if any) with the specified index.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool GetAnchorPos (
    int anchorIdx,
    ref Point point
)
```

Visual Basic Copy Code

```
Public Overridable Function GetAnchorPos( _
    anchorIdx As Integer, _
    ByRef point As Point _
) As Boolean
```

Parameters

anchorIdx

point

Return Value

See Also

GetAnchorPos Method Overload List
ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.GetEndPoint Method

Calculates the coordinates of the this connection point expressed in document coordinates.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Point GetEndPoint ()
```

Visual Basic Copy Code

```
Public Overridable Function GetEndPoint () As Point
```

Return Value

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.GetInitialPoint Method

Returns a point (usually the center of the bounding rectangle) contained within the associated node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Point GetInitialPoint ()
```

Visual Basic Copy Code

```
Public Overridable Function GetInitialPoint () As Point
```

Return Value

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.GetIntersection Method

Calculates the intersection point between the specified line segment and the contour of the node associated with this connection point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Point GetIntersection (
    Point pt1,
    Point pt2
)
```

Visual Basic Copy Code

```
Public Overridable Function GetIntersection( _
    pt1 As Point, _
```

```
    pt2 As Point _
) As Point
```

Parameters

pt1

pt2

Return Value

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.GetNearestBorderPoint Method

Returns the nearest point lying on the node border.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual Point GetNearestBorderPoint (
    Point point
)
```

Visual Basic Copy Code

```
Protected Overridable Function GetNearestBorderPoint( _
    point As Point _
) As Point
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying the point to align to node's borders.

Return Value

A **Microsoft.Maui.Graphics.Point** containing coordinates of nearest border point.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.GetNodeRect Method

Returns the bounding rectangle of the node associated with this connection point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Rect GetNodeRect (
    bool rotated
)
```

Visual Basic

```
Public Overridable Function GetNodeRect( _
    rotated As Boolean _
) As Rect
```

Parameters

rotated

Return Value

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.LinkChanges Method

Checks whether the relationship represented by the associated link would change if the link is attached to the specified node at the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public virtual bool LinkChanges (
    DiagramItem item,
    Point point
)
```

Visual Basic

```
Public Overridable Function LinkChanges( _
    item As DiagramItem, _
    point As Point _
) As Boolean
```

Parameters

item

The item under the mouse pointer.

point

The current mouse pointer position.

Return Value

true if the node should create a new ConnectionPoint instance, otherwise **false**.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.NearestAnchorPoint Method

Returns the anchor point closest to the specified point from the node's associated anchor pattern.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Point NearestAnchorPoint (
    Point point,
    ref int idx
)
```

Visual Basic Copy Code

```
Public Overridable Function NearestAnchorPoint( _
    point As Point, _
    ByRef idx As Integer _
) As Point
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** that should be aligned to anchor point.

idx
An integer variable that receives the anchor point index within AnchorPattern.

Return Value

A **Microsoft.Maui.Graphics.Point** containing anchor point coordinates.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.NodesIntersect Method

Checks whether the node associated with the current connection point and the node associated with the specified connection point intersect.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool NodesIntersect (
    ConnectionPoint connectionPoint
)
```

Visual Basic Copy Code

```
Public Function NodesIntersect( _
    connectionPoint As ConnectionPoint _
) As Boolean
```

Parameters

connectionPoint

Return Value

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.RemoveLinkFromNode Method

Removes the link associated with this connection point from the associated node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void RemoveLinkFromNode ()
```

Visual Basic Copy Code

```
Public Overridable Sub RemoveLinkFromNode ()
```

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.SameNode Method

Checks whether the current connection point and the specified connection point are associated with the same node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool SameNode (
    ConnectionPoint otherConnectionPoint
)
```

Visual Basic Copy Code

```
Public Overridable Function SameNode( _
    otherConnectionPoint As ConnectionPoint _
) As Boolean
```

Parameters

otherConnectionPoint

Return Value

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.SaveEndRelative Method

Updates the coordinates of the connection point represented by this link, relative to the bounding rectangle of the associated node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void SaveEndRelative ()
```

Visual Basic

```
Public Overridable Sub SaveEndRelative ()
```

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.UpdateFromPoint Method

Updates the connection state according to the specified point, without changing the related node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected virtual void UpdateFromPoint (
    Point point
)
```

Visual Basic

```
Protected Overridable Sub UpdateFromPoint( _
    point As Point _
)
```

Parameters

point
The current mouse pointer position.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint Properties

ConnectionPoint.Incoming Property

Gets a flag indicating whether this connection point is at the origin or the destination of the associated link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected bool Incoming { get; }
```

Visual Basic

```
Protected ReadOnly Property Incoming As Boolean
```

Property Value

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.Link Property

Gets the link associated with this connection point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected DiagramLink Link { get; }
```

Visual Basic Copy Code

```
Protected ReadOnly Property Link As DiagramLink
```

Property Value

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.Node Property

Gets the node associated with this connection point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual DiagramNode Node { get; }
```

Visual Basic Copy Code

```
Protected Overridable ReadOnly Property Node As DiagramNode
```

Property Value

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.RelativePosition Property

Gets or sets the position of this connection point, relative to the associated node's bounding rectangle, expressed as percentage.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point RelativePosition { get; set; }
```

Visual Basic Copy Code

```
Public Property RelativePosition As Point
```

Property Value

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

ConnectionPoint.Row Property

Gets the index of a constituent item within the associated node where the link represented by this connection point is connected, or -1, if the link is connected to the node itself.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual int Row { get; set; }
```

Visual Basic Copy Code

```
Public Overridable Property Row As Integer
```

Property Value

An integer ID of the related item, or -1 if the link connects to the node as integral entity.

See Also

ConnectionPoint Members
ConnectionPoint Class
MindFusion.Diagramming Namespace

Constants Class

Contains various global constants, also provides measure unit independent constants.

Syntax

C# Copy Code

```
public sealed class Constants
```

Visual Basic Copy Code

```
Public NotInheritable Class Constants
```

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.Constants

See Also

Constants Members
MindFusion.Diagramming Namespace

Constants Members

Contains various global constants, also provides measure unit independent constants.

The following tables list the members exposed by the Constants type.

Public Fields

	Name	Description
	E	A very small positive real number (0.01).
	IdentityMatrix	The identity matrix.
	NoAnchorPattern	A constant indicating no anchor pattern during binary serialization.

Public Methods

	Name	Description
	GetAutoScroll	Gets the number of units to scroll the control during auto-scroll operation.
	GetContainerFolderPadding	Gets the padding around the fold button of a container node.
	GetExpandIconSize	Gets the size of the expand/collapse manipulator icon.
	GetFitTextStep	Gets a value used when fitting the size of a node to its contained text.
	GetHalfInch	Gets the size of a half inch.
	GetHScrollButtonSize	Gets the size of horizontal scrollbar arrow buttons.
	GetInch	Gets the size of one inch.
	GetInflation	Gets a relax value used when testing object intersections.
	GetLineHitTest	Gets a value specifying relative distance from an arrow used during hit-testing operations.
	GetMarkSize	Gets the size of an anchor point mark.
	GetMillimeter	Gets the size of one millimeter.
	GetMinArrowheadSize	Gets the minimum allowed size of an arrowhead.
	GetMinItemSize	Gets the minimum size for a diagram node that can be achieved interactively.
	GetPixel	Gets the size of one pixel.
	GetReflHth	Gets the initial distance between adjacent control points of a reflexive arrow along the Y-axis.
	GetReflWth2	Gets the initial distance between adjacent control points of a reflexive link along the X-axis.
	GetRoundRectArc	Gets the radius of the arc of the rounded rectangle node shape.
	GetRouteGridSize	Gets the default size of the link routing grid.
	GetScrollerWdt	Gets the width of the scroller manipulator icon.
	GetSelPtTest	Gets a relax value used during selection handles hit-testing.

	GetTableHeaderOffset	Gets the width of a header row icon.
	GetVScrollButtonSize	Gets the size of vertical scrollbar arrow buttons.
	TouchHitDistance	Gets default value for unset TouchHitDistance.

See Also

Constants Class
MindFusion.Diagramming Namespace

Constants Fields

Constants.E Field

A very small positive real number (0.01).

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double E
```

Visual Basic Copy Code

```
Public Shared E As Double
```

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.IdentityMatrix Field

The identity matrix.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static IMatrix IdentityMatrix
```

Visual Basic Copy Code

```
Public Shared IdentityMatrix As IMatrix
```

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.NoAnchorPattern Field

A constant indicating no anchor pattern during binary serialization.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static int NoAnchorPattern
```

Visual Basic Copy Code

```
Public Shared NoAnchorPattern As Integer
```

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants Methods

Constants.GetAutoScroll Method

Gets the number of units to scroll the control during auto-scroll operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetAutoScroll (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetAutoScroll( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

A MeasureUnit instance.

Return Value

The number of units to add to or remove from the scroll position.

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetContainerFolderPadding Method

Gets the padding around the fold button of a container node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetContainerFolderPadding (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetContainerFolderPadding( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

A MeasureUnit instance.

Return Value

The padding space to leave around fold buttons, expressed in the specified unit.

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetExpandIconSize Method

Gets the size of the expand/collapse manipulator icon.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetExpandIconSize (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetExpandIconSize( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

A MeasureUnit instance.

Return Value

The +/- icon size, expressed in the specified unit.

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetFitTextStep Method

Gets a value used when fitting the size of a node to its contained text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetFitTextStep (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetFitTextStep( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit
A MeasureUnit instance.

Return Value

The step size, expressed in the specified unit.

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetHalfInch Method

Gets the size of a half inch.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetHalfInch (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetHalfInch( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit
A MeasureUnit instance.

Return Value

Half of an inch in the specified unit.

See Also

Constants.GetHScrollButtonSize Method

Gets the size of horizontal scrollbar arrow buttons.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Size GetHScrollButtonSize (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetHScrollButtonSize( _
    currUnit As MeasureUnit _
) As Size
```

Parameters

currUnit
A MeasureUnit instance.

Return Value

The size of scrollbar left and right arrow buttons, expressed in the specified unit.

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetInch Method

Gets the size of one inch.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetInch (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetInch( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit
A MeasureUnit instance.

Return Value

The size of an inch measured in the specified unit.

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetInflation Method

Gets a relax value used when testing object intersections.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetInflation (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetInflation( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetLineHitTest Method

Gets a value specifying relative distance from an arrow used during hit-testing operations.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetLineHitTest (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetLineHitTest( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetMarkSize Method

Gets the size of an anchor point mark.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetMarkSize (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetMarkSize( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetMillimeter Method

Gets the size of one millimeter.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetMillimeter (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetMillimeter( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit
A MeasureUnit instance.

Return Value

The size of a millimeter measured in the specified unit.

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetMinArrowheadSize Method

Gets the minimum allowed size of an arrowhead.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetMinArrowheadSize (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetMinArrowheadSize( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetMinItemSize Method

Gets the minimum size for a diagram node that can be achieved interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetMinItemSize (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetMinItemSize( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetPixel Method

Gets the size of one pixel.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetPixel (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetPixel( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetReflHth Method

Gets the initial distance between adjacent control points of a reflexive arrow along the Y-axis.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetReflHth (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetReflHth( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetReflWth2 Method

Gets the initial distance between adjacent control points of a reflexive link along the X-axis.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetReflWth2 (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetReflWth2( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetRoundRectArc Method

Gets the radius of the arc of the rounded rectangle node shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetRoundRectArc (
    MeasureUnit currUnit,
    double factor
)
```

Visual Basic Copy Code

```
Public Shared Function GetRoundRectArc( _
    currUnit As MeasureUnit, _
    factor As Double _
) As Double
```

Parameters

currUnit

factor

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetRouteGridSize Method

Gets the default size of the link routing grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetRouteGridSize (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetRouteGridSize( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetScrollerWdt Method

Gets the width of the scroller manipulator icon.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetScrollerWdt (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetScrollerWdt( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetSelPtTest Method

Gets a relax value used during selection handles hit-testing.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetSelPtTest (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetSelPtTest( _
    currUnit As MeasureUnit _
) As Double
```

Parameters

currUnit

Return Value

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetTableHeaderOffset Method

Gets the width of a header row icon.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double GetTableHeaderOffset (
    MeasureUnit unit
)
```

Visual Basic Copy Code

```
Public Shared Function GetTableHeaderOffset( _
    unit As MeasureUnit _
) As Double
```

Parameters

unit
A MeasureUnit instance.

Return Value

The offset of header rows, expressed in the specified unit.

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.GetVScrollButtonSize Method

Gets the size of vertical scrollbar arrow buttons.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Size GetVScrollButtonSize (
    MeasureUnit currUnit
)
```

Visual Basic Copy Code

```
Public Shared Function GetVScrollButtonSize( _
    currUnit As MeasureUnit _
) As Size
```

Parameters

currUnit
A MeasureUnit instance.

Return Value

The size of scrollbar top and bottom arrow buttons, expressed in the specified unit.

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

Constants.TouchHitDistance Method

Gets default value for unset TouchHitDistance.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double TouchHitDistance (
    MeasureUnit unit
)
```

Visual Basic Copy Code

```
Public Shared Function TouchHitDistance( _
    unit As MeasureUnit _
) As Double
```

Parameters

unit
Current measure unit.

Return Value

Default hit distance.

See Also

Constants Members
Constants Class
MindFusion.Diagramming Namespace

ContainerChildEventArgs Class

Contains arguments passed to events related to container children.

Syntax

C# Copy Code

```
public class ContainerChildEventArgs : NodeEventArgs
```

Visual Basic Copy Code

```
Public Class ContainerChildEventArgs
    Inherits NodeEventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.NodeEventArgs
      MindFusion.Diagramming.ContainerChildEventArgs

See Also

ContainerChildEventArgs Members
MindFusion.Diagramming Namespace

ContainerChildEventArgs Members

Contains arguments passed to events related to container children.

The following tables list the members exposed by the ContainerChildEventArgs type.

Public Constructors

	Name	Description
	ContainerChildEventArgs	Initializes a new instance of the **ContainerChildEventArgs** class.

Public Properties

	Name	Description
	AdjustmentHandle	Gets or sets the adjustment handle being moved by the user. (Inherited from NodeEventArgs.)
	Container	Refers to the container related to the event.
	Label	Gets or sets a NodeLabel related to this event. (Inherited from NodeEventArgs.)
	MouseButton	Gets which mouse button has been pressed. (Inherited from NodeEventArgs.)
	MousePosition	Gets or sets the current mouse pointer position. (Inherited from NodeEventArgs.)

	Node	Gets the node for which the event was raised. (Inherited from NodeEventArgs.)

Public Methods

	Name	Description
	ToString	NodeEventArgs.ToString override.

See Also

ContainerChildEventArgs Class
MindFusion.Diagramming Namespace

ContainerChildEventArgs Constructor

Initializes a new instance of the ContainerChildEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerChildEventArgs (
    ContainerNode container,
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public New ( _
    container As ContainerNode, _
    node As DiagramNode _
)
```

Parameters

container
A ContainerNode object related to the event.

node
A DiagramNode added to or removed from the container.

See Also

ContainerChildEventArgs Class
MindFusion.Diagramming Namespace

ContainerChildEventArgs Methods

ContainerChildEventArgs.ToString Method

NodeEventArgs.ToString override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override string ToString ()
```

Visual Basic Copy Code

```
Public Overrides Function ToString () As String
```

Return Value

See Also

ContainerChildEventArgs Members
ContainerChildEventArgs Class
MindFusion.Diagramming Namespace

ContainerChildEventArgs Properties

ContainerChildEventArgs.Container Property

Refers to the container related to the event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNode Container { get; set; }
```

Visual Basic Copy Code

```
Public Property Container As ContainerNode
```

Property Value

An instance of the ContainerNode class.

See Also

ContainerChildEventArgs Members
ContainerChildEventArgs Class
MindFusion.Diagramming Namespace

ContainerNode Class

Represents a diagram node that can contain other nodes.

Syntax

C# Copy Code

```
public class ContainerNode : DiagramNode, Scrollable
```

Visual Basic Copy Code

```
Public Class ContainerNode
    Inherits DiagramNode
    Implements Scrollable
```

Remarks

Users can drag nodes into a container interactively. When a dragged node is over a container, the container frame is painted using the HighlightPen property to indicate that the node would be added to the container.

Call the Add method to add a node to container programmatically, and the Remove method to remove a node from the container. Child nodes of a container are stored in its Children collection. The Container property of child nodes returns their

parent **ContainerNode**.

AutoShrink specifies whether the size of a container should shrink when a node is removed from it. The container node will not shrink to a size smaller than MinimumSize, even if it does not contain any items. Adding child nodes to a container will automatically grow its size, unless AutoGrow is set to **false**. Enable the Scrollable property to let users scroll container's content if it overflows.

A container node can be folded and unfolded, either interactively or programmatically, as specified through the Foldable and Folded properties. When a container is folded, it occupies a minimal area and its child nodes are hidden. You can get the original size of a folded container using the UnfoldedSize property.

The Caption property specifies the caption text of a container node, and CaptionBrush specifies its color. Use the CaptionFormat property to define the alignment and position of the caption text. The height of the caption bar of a container can be set using the CaptionHeight property.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.ContainerNode
        MindFusion.Diagramming.PagedContainerNode
        MindFusion.Diagramming.PagedContainerNode

See Also

ContainerNode Members
MindFusion.Diagramming Namespace

ContainerNode Members

Represents a diagram node that can contain other nodes.

The following tables list the members exposed by the ContainerNode type.

Public Constructors

	Name	Description
	ContainerNode	Overloaded. Initializes a new instance of the **ContainerNode** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowAddChildren	Gets or sets a value indicating whether child nodes can be added interactively to this container.
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AllowRemoveChildren	Gets or sets a value indicating whether child nodes can be removed interactively from this container.
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	AutoDeleteChildren	Gets or sets whether deleting this container should also remove its child items from the diagram.
	AutoGrow	Gets or sets a value indicating whether the container should grow in size when nodes are added to it.

	AutoShrink	Gets or sets a value indicating whether the container should shrink in size when nodes are removed from it.
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Caption	Gets or sets the text displayed as caption of the table.
	CaptionBackBrush	Gets or sets the brush used to paint the caption bar of this container.
	CaptionBrush	Gets or sets the brush used to paint the container caption.
	CaptionFormat	Gets or sets the caption alignment and format.
	CaptionHeight	Gets or sets the height of the container's caption bar.
	Children	Gets the child nodes of this container.
	ClipChildren	Gets or sets a value indicating whether child items should be clipped by container boundaries.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent **ContainerNode** containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveCaptionBackBrush	Gets the effective caption back brush of the container.
	EffectiveCaptionBrush	Gets the effective caption brush of the container.
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled.
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Foldable	Gets or sets a value indicating whether the user is allowed to fold this container.
	Folded	Gets or sets a value indicating whether this container is folded.
	FoldedSize	Gets or sets the size of the container when folded.
	FoldIconSize	Gets or sets the size of the fold/unfold icon.
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)

	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HighlightPen	Gets or sets the pen used to draw the container frame lines when an item is dragged over the container.
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	Margin	Gets or sets how much space should be left between contained nodes and the container borders.
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	MinimumSize	Gets or sets the minimum size of the container.
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Scrollable	Gets or sets a value indicating whether users are allowed to scroll the contents of this container.
	ScrollX	Gets or sets the container's horizontal scroll position.
	ScrollY	Gets or sets the container's vertical scroll position.
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)

	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Shape	Gets or sets the shape of the container's outline.
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	UnfoldedSize	Returns the original size of a folded container.
	Visible	Gets or sets a value indicating whether the container is visible.
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	Add	Adds a node to this container.
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	Arrange	Arranges the container's children with the given layout.

	ArrangeAnimated	Arranges the container's children using the given layout, and animates items moving to their new positions.
	ArrangeAsync	Arranges the container's children with the given layout asynchronously.
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	Clone	DiagramItem.Clone override. Creates a copy of this item.
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	ContainsRecursively	Determines whether this container contains the specified node, either directly or recursively through other containers.
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramNode.Draw override. Draws the container on the specified IGraphics surface.
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	DiagramNode.DrawLocal override. Draws the container on the specified IGraphics surface.
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified **IGraphics** surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
S	GetContainer	Retrieves the container node (if any) the specified node belongs to.
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectiveCaptionBackBrush	Returns a Brush object representing the currently effective caption background of the container.

	GetEffectiveCaptionBrush	Returns a Brush object representing the currently effective caption brush of the container.
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramNode.GetRepaintRect override.
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	DiagramItem.HitTest override.
	HitTestHandle	DiagramNode.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point.
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node.
	OnDragOver	Invoked when a diagram item is being dragged over this node.
	OnDropOver	Invoked when a diagram item is dropped over this node.
	OnLoad	DiagramItem.OnLoad override.
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	Remove	Removes a node from this container.
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)

	Resize	Resizes the node. (Inherited from DiagramNode.)
	ResizeToFitChildren	Resizes the container making it big enough to contain its child nodes.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	UpdateBounds	Overloaded.
	VisitNodes	Visits the diagram nodes in reverse Z order.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override.
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)

	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object.
	LoadFromXml	Loads the item content from an XML element.
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified.
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramNode.OnRemove override.
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramNode.RestoreProperties override.
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramNode.SaveProperties override.
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object.
	SaveToXml	Saves the item content into an XML element.
	ScaleElements	DiagramNode.ScaleElements override.
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary.

		(Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode Constructor

Initializes a new instance of the ContainerNode class.

Overload List

Name	Description
ContainerNode ()	Initializes a new instance of the **ContainerNode** class.
ContainerNode (ContainerNode)	Initializes a new instance of the **ContainerNode** class using the specified node as a prototype.
ContainerNode (Diagram)	Initializes a new instance of the **ContainerNode** class with the default values of the specified diagram.

See Also

ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode Constructor ()

Initializes a new instance of the ContainerNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNode ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ContainerNode Constructor Overload List
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode Constructor (ContainerNode)

Initializes a new instance of the ContainerNode class using the specified node as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNode (
    ContainerNode prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As ContainerNode _
)
```

Parameters

prototype

See Also

ContainerNode Constructor Overload List
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode Constructor (Diagram)

Initializes a new instance of the ContainerNode class with the default values of the specified diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNode (
    Diagram parent
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram _
)
```

Parameters

parent

See Also

ContainerNode Constructor Overload List
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode Methods

ContainerNode.Add Method

Adds a node to this container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Add (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Sub Add( _
    node As DiagramNode _
)
```

Parameters

node

The DiagramNode that will be added to the container.

Remarks

By default, if container's bounding rectangle does not include specified node's bounds, the container grows automatically to accommodate the new node. To disable that, set AutoGrow to **false**.

Child nodes that have been added to a container are stored in its Children collection. The parent ContainerNode of a child node can be accessed via the Container property.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
Remove Method
ClipChildren Property

ContainerNode.Arrange Method

Arranges the container's children with the given layout.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Arrange (
    ILayout layout
)
```

Visual Basic Copy Code

```
Public Function Arrange( _
    layout As ILayout _
) As Boolean
```

Parameters

layout
An instance of Layout -derived class specifying the layout algorithm to apply.

Return Value

true if the container has been arranged successfully, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.ArrangeAnimated Method

Arranges the container's children using the given layout, and animates items moving to their new positions.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ArrangeAnimated (
    ILayout layout,
    int duration,
    AnimationType animationType,
    EasingType easingType
)
```

Visual Basic Copy Code

```
Public Function ArrangeAnimated( _
    layout As ILayout, _
    duration As Integer, _
    animationType As AnimationType, _
    easingType As EasingType _
) As Boolean
```

Parameters

layout
An instance of Layout -derived class specifying the layout algorithm to apply.

duration
An integer, specifying the duration of the animation in milliseconds.

animationType
A member of the AnimationType enumeration, specifying the animation type to use.

easingType
A member of the EasingType enumeration, specifying the type of easing function to apply to the animation.

Return Value

true if the container has been arranged successfully, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.ArrangeAsync Method

Arranges the container's children with the given layout asynchronously.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task<Boolean> ArrangeAsync (
    ILayout layout
)
```

Visual Basic Copy Code

```
Public Function ArrangeAsync( _
    layout As ILayout _
) As Task(Of Boolean)
```

Parameters

layout
An instance of Layout -derived class specifying the layout algorithm to apply.

Return Value

true if the container has been arranged successfully, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.Clone Method

DiagramItem.Clone override. Creates a copy of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override DiagramItem Clone (
    bool clipboard
)
```

Visual Basic Copy Code

```
Public Overrides Function Clone( _
    clipboard As Boolean _
) As DiagramItem
```

Parameters

clipboard
true if copying during clipboard operations, or **false** otherwise.

Return Value

A copy of this ContainerNode.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.ContainsRecursively Method

Determines whether this container contains the specified node, either directly or recursively through other containers.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ContainsRecursively (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Function ContainsRecursively( _
    node As DiagramNode _
) As Boolean
```

Parameters

node
The DiagramNode to check.

Return Value

true if this container contains the node; otherwise, **false**.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.CreateProperties Method

DiagramItem.CreateProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemProperties CreateProperties ()
```

Visual Basic Copy Code

```
Protected Overrides Function CreateProperties () As DiagramItemProperties
```

Return Value

An instance of the DiagramItemProperties class.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.Draw Method

DiagramNode.Draw override. Draws the container on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Draw (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub Draw( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics

The IGraphics surface to draw the node on.

options

Provides options for the drawing operation.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.DrawLocal Method

DiagramNode.DrawLocal override. Draws the container on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawLocal (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawLocal( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics

An IGraphics object where the node should be rendered.

options

Provides options for the drawing operation.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.GetContainer Method

Retrieves the container node (if any) the specified node belongs to.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static ContainerNode GetContainer (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Shared Function GetContainer( _
    node As DiagramNode _
) As ContainerNode
```

Parameters

node

The DiagramNode whose container you need to find.

Return Value

A ContainerNode that contains the specified node, or **null** (**Nothing** in Visual Basic) if the node is not contained.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.GetEffectiveCaptionBackBrush Method

Returns a Brush object representing the currently effective caption background of the container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush GetEffectiveCaptionBackBrush (
    Rect bounds,
    double dx,
```

```
    double dy
)
```

Visual Basic Copy Code

```
Public Function GetEffectiveCaptionBackBrush( _
    bounds As Rect, _
    dx As Double, _
    dy As Double _
) As Brush
```

Parameters

bounds

dx

dy

Return Value

Remarks

This method returns either the container's CaptionBackBrush if not null, or the CaptionBackBrush property of the container's style. This method can return null.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.GetEffectiveCaptionBrush Method

Returns a Brush object representing the currently effective caption brush of the container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush GetEffectiveCaptionBrush ()
```

Visual Basic Copy Code

```
Public Function GetEffectiveCaptionBrush () As Brush
```

Return Value

A Brush instance.

Remarks

This method returns either the container's CaptionBrush if not **null**, or the TextBrush property of the container's style.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.GetRepaintRect Method

DiagramNode.GetRepaintRect override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Rect GetRepaintRect (
    bool includeConnected
)
```

Visual Basic Copy Code

```
Public Overrides Function GetRepaintRect( _
    includeConnected As Boolean _
) As Rect
```

Parameters

includeConnected

true is related items should be considered too, or **false** otherwise.

Return Value

A rectangular region that should be repainted when this node is modified.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.HitTest Method

DiagramItem.HitTest override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override DiagramItem HitTest (
    Point point,
    double threshold,
    bool onlyNodes,
    bool onlyInteractive,
    bool includeHandles,
    bool includeLabels
)
```

Visual Basic Copy Code

```
Public Overrides Function HitTest( _
    point As Point, _
    threshold As Double, _
    onlyNodes As Boolean, _
    onlyInteractive As Boolean, _
    includeHandles As Boolean, _
    includeLabels As Boolean _
) As DiagramItem
```

Parameters

point

A **Microsoft.Maui.Graphics.Point** value specifying the logical coordinates of a diagram point.

threshold

Specifies the size of the area around the item outlines where the point is still considered inside the item.

onlyNodes

true to exclude child links from the search.

onlyInteractive

true to exclude locked items from the search.

includeHandles
true to also consider adjustment handle locations (possibly outside of node's bounds).

includeLabels
true to also consider item label locations (possibly outside of node's bounds).

Return Value

The item that contains the specified point.

Remarks

This method performs recursive search if items contain child items.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.HitTestHandle Method

DiagramNode.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override AdjustmentHandle HitTestHandle (
    Point point,
    double distance
)
```

Visual Basic Copy Code

```
Public Overrides Function HitTestHandle( _
    point As Point, _
    distance As Double _
) As AdjustmentHandle
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** value specifying where to look for a selection handle.

distance
Maximum distance from a handle at which it is still considered as hit.

Return Value

An AdjustmentHandle instance representing the adjustment handle found at the specified location, or **null** if there is none.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.LoadFromJson Method

Loads the node from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the node's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.LoadFromXml Method

Loads the item content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** containing the item's data.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.OnChildModified Method

Called when a group child node has been modified.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnChildModified (
    DiagramNode node,
    AdjustmentHandles handle
)
```

Visual Basic Copy Code

```
Protected Overrides Sub OnChildModified( _
    node As DiagramNode, _
    handle As AdjustmentHandles _
)
```

Parameters

node
The DiagramNode that has been moved or resized.

handle
The adjustment handle used to modify the node.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.OnDragOut Method

Invoked when a diagram item is being dragged out of this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void OnDragOut (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Overrides Sub OnDragOut( _
    item As DiagramItem _
)
```

Parameters

item
The DiagramItem being dragged.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.OnDragOver Method

Invoked when a diagram item is being dragged over this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool OnDragOver (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Overrides Function OnDragOver( _
    item As DiagramItem _
) As Boolean
```

Parameters

item
The DiagramItem being dragged.

Return Value

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.OnDropOver Method

Invoked when a diagram item is dropped over this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool OnDropOver (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Overrides Function OnDropOver( _
    item As DiagramItem _
) As Boolean
```

Parameters

item
The DiagramItem being dragged.

Return Value

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.OnLoad Method

DiagramItem.OnLoad override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void OnLoad ()
```

Visual Basic Copy Code

```
Public Overrides Sub OnLoad ()
```

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.OnRemove Method

DiagramNode.OnRemove override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnRemove ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnRemove ()
```

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.Remove Method

Removes a node from this container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Remove (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Sub Remove( _
    node As DiagramNode _
)
```

Parameters

node

The DiagramNode that should be removed from the container.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
Add Method

ContainerNode.ResizeToFitChildren Method

Resizes the container making it big enough to contain its child nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ResizeToFitChildren (
    bool allowShrink,
    Size margins
)
```

Visual Basic Copy Code

```
Public Sub ResizeToFitChildren( _
    allowShrink As Boolean, _
    margins As Size _
)
```

Parameters

allowShrink

true to allow reducing container's size, or **false** otherwise.

margins

A **Microsoft.Maui.Graphics.Size** specifying size of margin space left around container's child items.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
Children Property
AutoGrow Property
AutoShrink Property

ContainerNode.RestoreProperties Method

DiagramNode.RestoreProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties -derived object containing a snapshot of the item's properties.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.SaveProperties Method

DiagramNode.SaveProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties -derived object where the item's properties should be saved.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.SaveToJson Method

Saves the node as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
```

```
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject instance to contain the serialized data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.SaveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** the item's data should be stored into.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.ScaleElements Method

DiagramNode.ScaleElements override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void ScaleElements (
    double scale
)
```

Visual Basic Copy Code

```
Protected Overrides Sub ScaleElements( _
    scale As Double _
)
```

Parameters

scale
The value by which to multiply node elements' sizes.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.UpdateBounds Method

Overload List

Name	Description
ContainerNode.UpdateBounds ()	Updates the container bounds to fit its contents.
ContainerNode.UpdateBounds (Boolean)	Updates the container bounds to fit its contents.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.UpdateBounds Method ()

Updates the container bounds to fit its contents.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void UpdateBounds ()
```

Visual Basic Copy Code

```
Public Sub UpdateBounds ()
```

See Also

UpdateBounds Method Overload List
ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.UpdateBounds Method (Boolean)

Updates the container bounds to fit its contents.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void UpdateBounds (
    bool includeLinks
)
```

Visual Basic Copy Code

```
Public Sub UpdateBounds( _
    includeLinks As Boolean _
)
```

Parameters

includeLinks
Specifies whether the bounding rectangle should contain the control points of links whose ends are both child nodes of this container.

See Also

UpdateBounds Method Overload List
ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.VisitNodes Method

Visits the diagram nodes in reverse Z order.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool VisitNodes (
    Point point,
    VisitNode visitNode,
    DiagramItem relatedItem
)
```

Visual Basic Copy Code

```
Public Function VisitNodes( _
    point As Point, _
    visitNode As VisitNode, _
    relatedItem As DiagramItem _
) As Boolean
```

Parameters

point
Location of the nodes that should be visited.

visitNode
An operation to be performed on the nodes.

relatedItem
Parameter to pass to the visitNode operation.

Return Value

See Also

ContainerNode Properties

ContainerNode.AllowAddChildren Property

Gets or sets a value indicating whether child nodes can be added interactively to this container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowAddChildren { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowAddChildren As Boolean
```

Property Value

true to allow adding child nodes, otherwise **false**. The default value is **true**.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
AllowRemoveChildren Property

ContainerNode.AllowRemoveChildren Property

Gets or sets a value indicating whether child nodes can be removed interactively from this container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowRemoveChildren { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowRemoveChildren As Boolean
```

Property Value

true to allow removing child nodes, otherwise **false**. The default value is **true**.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
AllowAddChildren Property

ContainerNode.AutoDeleteChildren Property

Gets or sets whether deleting this container should also remove its child items from the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoDeleteChildren { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoDeleteChildren As Boolean
```

Property Value

true to automatically delete child items, or **false** otherwise.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
Children Property

ContainerNode.AutoGrow Property

Gets or sets a value indicating whether the container should grow in size when nodes are added to it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoGrow { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoGrow As Boolean
```

Property Value

true to grow container's size automatically, or **false** otherwise. The default value is **true**.

Remarks

When **AutoGrow** is enabled, the container size will grow to include newly added nodes when they wouldn't fit in its current size. The new **Bounds** of the container is set to the union of bounding rectangles of all child nodes inflated by Margin.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
AutoShrink Property
ResizeToFitChildren Method

ContainerNode.AutoShrink Property

Gets or sets a value indicating whether the container should shrink in size when nodes are removed from it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoShrink { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoShrink As Boolean
```

Property Value

true if the container should shrink automatically; otherwise, **false**. The default is **false**.

Remarks

The container won't shrink to a size smaller than MinimumSize.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
AutoGrow Property
ResizeToFitChildren Method

ContainerNode.Caption Property

Gets or sets the text displayed as caption of the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Caption { get; set; }
```

Visual Basic Copy Code

```
Public Property Caption As String
```

Property Value

A string containing the container's caption text.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
CaptionBrush Property
CaptionFormat Property
CaptionHeight Property

ContainerNode.CaptionBackBrush Property

Gets or sets the brush used to paint the caption bar of this container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush CaptionBackBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionBackBrush As Brush
```

Property Value

An instance of Brush -derived class.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
CaptionHeight Property
CaptionBrush Property
DividerPen Property

ContainerNode.CaptionBrush Property

Gets or sets the brush used to paint the container caption.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush CaptionBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
Caption Property
CaptionBackBrush Property
CaptionHeight Property
CaptionFormat Property

ContainerNode.CaptionFormat Property

Gets or sets the caption alignment and format.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringFormat CaptionFormat { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionFormat As StringFormat
```

Property Value

An instance of the StringFormat class.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
Caption Property
CaptionHeight Property

ContainerNode.CaptionHeight Property

Gets or sets the height of the container's caption bar.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double CaptionHeight { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionHeight As Double
```

Property Value

A **double** value specifying the height of the caption bar. Initialized with the value of ContainerCaptionHeight.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
CaptionFormat Property
Caption Property
CaptionBackBrush Property

ContainerNode.Children Property

Gets the child nodes of this container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ReadOnlyDiagramNodeCollection Children { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Children As ReadOnlyDiagramNodeCollection
```

Property Value

A collection containing the child nodes of this container.

Remarks

Call the Add method of ContainerNode to add a child node to the container. Call Remove to remove a child node.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
Container Property
AutoDeleteChildren Property
ResizeToFitChildren Method

ContainerNode.ClipChildren Property

Gets or sets a value indicating whether child items should be clipped by container boundaries.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ClipChildren { get; set; }
```

Visual Basic Copy Code

```
Public Property ClipChildren As Boolean
```

Property Value

true to clip child items; otherwise, **false**. The default value is **true**.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
Shape Property

ContainerNode.EffectiveCaptionBackBrush Property

Gets the effective caption back brush of the container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush EffectiveCaptionBackBrush { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectiveCaptionBackBrush As Brush
```

Property Value

An instance of Brush -derived class.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.EffectiveCaptionBrush Property

Gets the effective caption brush of the container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush EffectiveCaptionBrush { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectiveCaptionBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.EnableStyledText Property

Gets or sets a value indicating whether styled text rendering is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableStyledText { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableStyledText As Boolean
```

Property Value

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.Foldable Property

Gets or sets a value indicating whether the user is allowed to fold this container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Foldable { get; set; }
```

Visual Basic Copy Code

```
Public Property Foldable As Boolean
```

Property Value

true if the container can be folded, or **false** otherwise. Initialized with the value of ContainersFoldable.

Remarks

A foldable container can be folded and unfolded by clicking the up/down arrow displayed in the caption bar. The appearance of that icon can be customized by setting **Renderers.FoldIconRenderer**.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
Folded Property

ContainerNode.Folded Property

Gets or sets a value indicating whether this container is folded.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Folded { get; set; }
```

Visual Basic Copy Code

```
Public Property Folded As Boolean
```

Property Value

true if the container is folded; otherwise, **false**. The default is **false**.

Remarks

When folded, the container children are not visible, and only its caption bar is displayed.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
Foldable Property
Diagram.FoldedContainerSize Property

ContainerNode.FoldedSize Property

Gets or sets the size of the container when folded.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size FoldedSize { get; set; }
```

Visual Basic Copy Code

```
Public Property FoldedSize As Size
```

Property Value

A **Size** instance specifying the folded size of a container. The default value is **Empty**.

Remarks

If this property is not set, the width of the folded container is set to its MinimumSize.Width, and the height to CaptionHeight.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.FoldIconSize Property

Gets or sets the size of the fold/unfold icon.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Nullable<Double> FoldIconSize { get; set; }
```

Visual Basic Copy Code

```
Public Property FoldIconSize As Nullable(Of Double)
```

Property Value

An optional **double** value specifying the size of fold/unfold icon. The default value is **null**.

Remarks

If this property is not set, the fold icon derives its size from CaptionHeight.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
Foldable Property

ContainerNode.HighlightPen Property

Gets or sets the pen used to draw the container frame lines when an item is dragged over the container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen HighlightPen { get; set; }
```

Visual Basic Copy Code

```
Public Property HighlightPen As Pen
```

Property Value

An instance of the Pen class. Initialized with the value of ContainerHighlightPen.

Remarks

The highlight rectangle is not drawn when AllowAddChildren is disabled.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.Margin Property

Gets or sets how much space should be left between contained nodes and the container borders.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Margin { get; set; }
```

Visual Basic Copy Code

```
Public Property Margin As Double
```

Property Value

A float value specifying the margin size. Initialized with the value of ContainerMargin.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
AutoShrink Property

ContainerNode.MinimumSize Property

Gets or sets the minimum size of the container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size MinimumSize { get; set; }
```

Visual Basic Copy Code

```
Public Property MinimumSize As Size
```

Property Value

A **Size** object specifying the minimum size. Initialized with the value of ContainerMinimumSize.

Remarks

The container won't shrink smaller than that size, even if it does not contain any items.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
AutoShrink Property

ContainerNode.Scrollable Property

Gets or sets a value indicating whether users are allowed to scroll the contents of this container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Scrollable { get; set; }
```

Visual Basic Copy Code

```
Public Property Scrollable As Boolean
```

Property Value

true to let users scroll this container, or **false** otherwise.

Remarks

Note that in order to let container's contents overflow and become scrollable, you must either disable AutoGrow, or allow users reduce container size by changing HandlesStyle and EnabledHandles properties from their default MoveOnly / Move values.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
ScrollX Property
ScrollY Property

ContainerNode.ScrollX Property

Gets or sets the container's horizontal scroll position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ScrollX { get; set; }
```

Visual Basic Copy Code

```
Public Property ScrollX As Double
```

Property Value

A **float** value specifying horizontal scroll offset.

Remarks

ScrollX is one of the few properties that automatically create a record in the undo history. If you need to undo a sequence of property changes that includes **ScrollX** as a single operation, you will need to use composite undo commands.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
ScrollY Property
Scrollable Property

ContainerNode.ScrollY Property

Gets or sets the container's vertical scroll position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ScrollY { get; set; }
```

Visual Basic Copy Code

```
Public Property ScrollY As Double
```

Property Value

A **float** value specifying vertical scroll offset.

Remarks

ScrollY is one of the few properties that automatically create a record in the undo history. If you need to undo a sequence of property changes that includes **ScrollY** as a single operation, you will need to use composite undo commands.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
ScrollX Property
Scrollable Property

ContainerNode.Shape Property

Gets or sets the shape of the container's outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SimpleShape Shape { get; set; }
```

Visual Basic Copy Code

```
Public Property Shape As SimpleShape
```

Property Value

A member of the SimpleShape enumeration.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace
ClipChildren Property

ContainerNode.UnfoldedSize Property

Returns the original size of a folded container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size UnfoldedSize { get; set; }
```

Visual Basic Copy Code

```
Public Property UnfoldedSize As Size
```

Property Value

A **Size** instance containing the original size of the container.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNode.Visible Property

Gets or sets a value indicating whether the container is visible.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool Visible { get; set; }
```

Visual Basic Copy Code

```
Public Overrides Property Visible As Boolean
```

Property Value

true if the item is visible; otherwise, **false**.

See Also

ContainerNode Members
ContainerNode Class
MindFusion.Diagramming Namespace

ContainerNodeProperties Class

Stores the values of ContainerNode properties during undo / redo operations.

Syntax

C# Copy Code

```
public class ContainerNodeProperties : DiagramNodeProperties
```

Visual Basic Copy Code

```
Public Class ContainerNodeProperties
    Inherits DiagramNodeProperties
```

Remarks

If you need to implement undo support for the additional properties defined by a class derived from ContainerNode, do so by deriving from **ContainerNodeProperties** and overriding the SaveProperties, RestoreProperties, and CreateProperties methods of

ContainerNode.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItemProperties
    MindFusion.Diagramming.DiagramNodeProperties
      MindFusion.Diagramming.ContainerNodeProperties

See Also

ContainerNodeProperties Members
MindFusion.Diagramming Namespace

ContainerNodeProperties Members

Stores the values of ContainerNode properties during undo / redo operations.

The following tables list the members exposed by the ContainerNodeProperties type.

Public Constructors

	Name	Description
	ContainerNodeProperties	Initializes a new instance of the **ContainerNodeProperties** class.

Public Fields

	Name	Description
	Caption	For internal use.
	CaptionHeight	For internal use.
	ItemClass	For internal use. (Inherited from DiagramItemProperties.)
	labels	For internal use. (Inherited from DiagramNodeProperties.)
	Margin	For internal use.
	Selected	For internal use. (Inherited from DiagramItemProperties.)
	Shape	For internal use.
	Text	For internal use. (Inherited from DiagramItemProperties.)
	TextPadding	For internal use. (Inherited from DiagramItemProperties.)
	ZIndex	For internal use. (Inherited from DiagramItemProperties.)

Public Methods

	Name	Description
	FreeResources	Frees stored resources which will no longer be used. (Inherited from DiagramItemProperties.)

See Also

ContainerNodeProperties Class
MindFusion.Diagramming Namespace

ContainerNodeProperties Fields

ContainerNodeProperties.Caption Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Caption
```

Visual Basic Copy Code

```
Public Caption As String
```

See Also

ContainerNodeProperties Members
ContainerNodeProperties Class
MindFusion.Diagramming Namespace

ContainerNodeProperties.CaptionHeight Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double CaptionHeight
```

Visual Basic Copy Code

```
Public CaptionHeight As Double
```

See Also

ContainerNodeProperties Members
ContainerNodeProperties Class
MindFusion.Diagramming Namespace

ContainerNodeProperties.Margin Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Margin
```

Visual Basic Copy Code

```
Public Margin As Double
```

See Also

ContainerNodeProperties Members
ContainerNodeProperties Class
MindFusion.Diagramming Namespace

ContainerNodeProperties.Shape Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SimpleShape Shape
```

Visual Basic Copy Code

```
Public Shape As SimpleShape
```

See Also

ContainerNodeProperties Members
ContainerNodeProperties Class
MindFusion.Diagramming Namespace

ContainerNodeProperties Constructor

Initializes a new instance of the ContainerNodeProperties class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNodeProperties ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ContainerNodeProperties Class
MindFusion.Diagramming Namespace

ContainerNodeStyle Class

Represents a reusable set of properties that define the appearance of container nodes.

Syntax

C# Copy Code

```
public class ContainerNodeStyle : DiagramNodeStyle
```

Visual Basic — Copy Code

```
Public Class ContainerNodeStyle
    Inherits DiagramNodeStyle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle
      MindFusion.Diagramming.DiagramNodeStyle
        MindFusion.Diagramming.ContainerNodeStyle

See Also

ContainerNodeStyle Members
MindFusion.Diagramming Namespace

ContainerNodeStyle Members

Represents a reusable set of properties that define the appearance of container nodes.

The following tables list the members exposed by the ContainerNodeStyle type.

Public Constructors

	Name	Description
	ContainerNodeStyle	Initializes a new instance of the **ContainerNodeStyle** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of the item. (Inherited from DiagramItemStyle.)
	Effects	Gets a list of the effects applied to the node. (Inherited from DiagramNodeStyle.)
	FontFamily	Gets or sets the font family of the item text. (Inherited from DiagramItemStyle.)
	FontSize	Gets or sets the font size of the item text. (Inherited from DiagramItemStyle.)
	FontStyle	Gets or sets the font style of the item text. (Inherited from DiagramItemStyle.)
	FontUnit	Gets or sets the font unit of the item text. (Inherited from DiagramItemStyle.)
	ShadowBrush	Gets or sets the brush used to paint the shadow of the item. (Inherited from DiagramItemStyle.)
	Stroke	Gets or sets the brush used to paint the outline of the item. (Inherited from DiagramItemStyle.)
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items. (Inherited from DiagramItemStyle.)
	StrokeThickness	Gets or sets the thickness of the item's outline, expressed in the current units of measure. (Inherited from DiagramItemStyle.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItemStyle.)

Public Methods

	Name	Description
	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
S	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JsonObject. (Inherited from DiagramNodeStyle.)
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)
	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a JsonObject. (Inherited from DiagramNodeStyle.)
	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

ContainerNodeStyle Class
MindFusion.Diagramming Namespace

ContainerNodeStyle Constructor

Initializes a new instance of the ContainerNodeStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNodeStyle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ContainerNodeStyle Class
MindFusion.Diagramming Namespace

ContainerNodeStyle Events

ContainerPage Class

Represents a page in a PagedContainerNode.

Syntax

C# Copy Code

```
public class ContainerPage
```

Visual Basic Copy Code

```
Public Class ContainerPage
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ContainerPage

See Also

ContainerPage Members
MindFusion.Diagramming Namespace

ContainerPage Members

Represents a page in a PagedContainerNode.

The following tables list the members exposed by the ContainerPage type.

Public Constructors

	Name	Description
	ContainerPage	Overloaded.

Public Properties

	Name	Description
	Links	Gets the links on this page.
	Nodes	Gets the nodes on this page.
	Title	Gets or sets the title of this page.

See Also

ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage Constructor

Overload List

Name	Description
ContainerPage ()	Initializes a new instance of the ContainerPage class.
ContainerPage (ContainerPage)	Initializes a new instance of the **ContainerPage** class.

See Also

ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage Constructor ()

Initializes a new instance of the ContainerPage class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerPage ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ContainerPage Constructor Overload List
ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage Constructor (ContainerPage)

Initializes a new instance of the ContainerPage class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerPage (
    ContainerPage prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As ContainerPage _
)
```

Parameters

prototype

See Also

ContainerPage Properties

ContainerPage.Links Property

Gets the links on this page.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection Links { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Links As DiagramLinkCollection
```

Property Value

A DiagramLinkCollection instance.

Remarks

Unlike Nodes, the control does not add links to this collection automatically, as ownership could be ambiguous (when considering nested containers, cross-container links, etc). This list lets you associate links with a page yourself when you want it hidden along with the page. The container automatically hides links between direct child nodes of non-current page.

See Also

ContainerPage Members
ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage.Nodes Property

Gets the nodes on this page.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeCollection Nodes { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Nodes As DiagramNodeCollection
```

Property Value

A DiagramNodeCollection instance.

Remarks

PagedContainerNode automatically adds child nodes to this list when users drag them into CurrentPage.

See Also

ContainerPage Members
ContainerPage Class
MindFusion.Diagramming Namespace

ContainerPage.Title Property

Gets or sets the title of this page.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Title { get; set; }
```

Visual Basic Copy Code

```
Public Property Title As String
```

Property Value

A string containing title text.

Remarks

PagedContainerNode set its Caption text to the title of CurrentPage.

See Also

ContainerPage Members
ContainerPage Class
MindFusion.Diagramming Namespace

ContainmentKeys Enumeration

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum ContainmentKeys
```

Visual Basic Copy Code

```
Public Enum ContainmentKeys
```

Members

	Member name	Description
	InContainer	Set for Group.Attachment objects for nodes in a container.

See Also

MindFusion.Diagramming Namespace

ControlMouseAction Enumeration

Specifies available ways to handle mouse clicks on .NET controls hosted in a **ControlNode**.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum ControlMouseAction
```

Visual Basic Copy Code

```
Public Enum ControlMouseAction
```

Members

	Member name	Description
	PassToControl	Let the control handle the mouse click. Do not select the host node.
	SelectNode	The host node gets selected when the control is clicked. The control is left to handle the mouse click too.
	IgnoreControl	MvcDiagram tries to intercept the mouse message and not pass it to the control. That allows drawing links from anywhere within the control.

Remarks

IgnoreControl works only with controls that do not capture the mouse.

See Also

MindFusion.Diagramming Namespace

ControlPointEventArgs Class

Contains arguments passed to handlers of the ShapeModified event.

Syntax

C# Copy Code

```
public class ControlPointEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class ControlPointEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.ControlPointEventArgs

See Also

ControlPointEventArgs Members
MindFusion.Diagramming Namespace

ControlPointEventArgs Members

Contains arguments passed to handlers of the ShapeModified event.

The following tables list the members exposed by the ControlPointEventArgs type.

Public Constructors

	Name	Description
	ControlPointEventArgs	Initializes a new instance of the **ControlPointEventArgs** class.

Public Properties

	Name	Description
	ControlPoint	Gets or sets the ShapeControlPoint used to modify the node's shape.
	NewValue	Gets or sets the new position of the control point.
	Node	Gets or sets the ShapeNode whose Shape has been modified.
	OldValue	Gets or sets the original position of the control point.

See Also

ControlPointEventArgs Class
MindFusion.Diagramming Namespace

ControlPointEventArgs Constructor

Initializes a new instance of the ControlPointEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ControlPointEventArgs (
    ShapeControlPoint controlPoint,
    ShapeNode node,
    ShapeFormulaParam oldValue,
    ShapeFormulaParam newValue
)
```

Visual Basic Copy Code

```
Public New ( _
    controlPoint As ShapeControlPoint, _
    node As ShapeNode, _
    oldValue As ShapeFormulaParam, _
    newValue As ShapeFormulaParam _
)
```

Parameters

controlPoint
The ShapeControlPoint used to modify the node's shape.

node
The ShapeNode whose Shape has been modified.

oldValue
A ShapeFormulaParam specifying the original position of the control point.

newValue
A **ShapeFormulaParam** specifying the new position of the control point.

See Also

ControlPointEventArgs Class
MindFusion.Diagramming Namespace

ControlPointEventArgs Properties

ControlPointEventArgs.ControlPoint Property

Gets or sets the ShapeControlPoint used to modify the node's shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeControlPoint ControlPoint { get; set; }
```

Visual Basic Copy Code

```
Public Property ControlPoint As ShapeControlPoint
```

Property Value

An instance of the ShapeControlPoint class.

See Also

ControlPointEventArgs Members
ControlPointEventArgs Class
MindFusion.Diagramming Namespace

ControlPointEventArgs.NewValue Property

Gets or sets the new position of the control point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeFormulaParam NewValue { get; set; }
```

Visual Basic Copy Code

```
Public Property NewValue As ShapeFormulaParam
```

Property Value

A ShapeFormulaParam instance specifying the new position of the shape control point.

See Also

ControlPointEventArgs Members
ControlPointEventArgs Class
MindFusion.Diagramming Namespace

ControlPointEventArgs.Node Property

Gets or sets the ShapeNode whose Shape has been modified.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNode Node { get; set; }
```

Visual Basic Copy Code

```
Public Property Node As ShapeNode
```

Property Value

An instance of the ShapeNode class.

See Also

ControlPointEventArgs Members
ControlPointEventArgs Class
MindFusion.Diagramming Namespace

ControlPointEventArgs.OldValue Property

Gets or sets the original position of the control point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeFormulaParam OldValue { get; set; }
```

Visual Basic Copy Code

```
Public Property OldValue As ShapeFormulaParam
```

Property Value

A ShapeFormulaParam instance specifying the original position of the shape control point.

See Also

ControlPointEventArgs Members
ControlPointEventArgs Class
MindFusion.Diagramming Namespace

Converter Class

Provides capabilities to convert various types to and from strings in a culture-independent way.

Syntax

C# Copy Code

```
public sealed class Converter
```

Visual Basic Copy Code

```
Public NotInheritable Class Converter
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Converter

See Also

Converter Members
MindFusion.Diagramming Namespace

Converter Members

Provides capabilities to convert various types to and from strings in a culture-independent way.

The following tables list the members exposed by the Converter type.

Public Methods

	Name	Description
	FromBoolean	Converts the specified bool value to string.
	FromByte	Converts the specified byte value to string.
	FromChar	Converts the specified char value to string.
	FromColor	Converts the specified color value to string in the format #AARRGGBB.
	FromDecimal	Converts the specified decimal to string.
	FromDouble	Converts the specified double value to string.
	FromEnum	Converts the specified enum value to a string.
	FromImage	Converts the specified image to string.
	FromInt16	Converts the specified 16-bit integer value to string.
	FromInt32	Converts the specified 32-bit integer value to string.
	FromInt64	Converts the specified 64-bit integer to string.
	FromSByte	Converts the specified 8-bit signed integer to string.
	FromSingle	Converts the specified float value to string.
	FromStream	Converts the specified stream to string.
	FromStreamV4	Converts the specified memory stream to string.
	FromUInt16	Converts the specified unsigned 16-bit integer value to string.
	FromUInt32	Converts the specified unsigned 32-bit integer value to string.
	FromUInt64	Converts the specified unsigned 64-bit integer value to string.
	ImageFromString	Converts the specified string value to image.
	ToBoolean	Converts the specified string value to bool.
	ToByte	Converts the specified string value to byte value.
	ToChar	Converts the specified string value to char.
	ToColor	Converts the specified string value to Color.
	ToDecimal	Converts the specified string value to decimal.

	ToDouble	Converts the specified string value to double.
	ToEnum	Converts the specified string value to its corresponding enumeration member of the specified type.
	ToImage	Converts the specified string value to image.
	ToInt16	Converts the specified string value to 16-bit integer.
	ToInt32	Overloaded. Converts a string value to 32-bit integer.
	ToInt64	Converts the specified string value to 64-bit integer.
	ToSByte	Converts the specified string value to 8-bit signed integer.
	ToSingle	Converts the specified string value to float.
	ToStream	Converts the specified string value to stream.
	ToStreamV4	Converts the specified string value to stream.
	ToUint16	Converts the specified string value to unsigned 16-bit integer.
	ToUint32	Converts the specified string value to unsigned 32-bit integer.
	ToUint64	Converts the specified string value to unsigned 64-bit integer.

See Also

Converter Class
MindFusion.Diagramming Namespace

Converter Methods

Converter.FromBoolean Method

Converts the specified bool value to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromBoolean (
    bool val
)
```

Visual Basic Copy Code

```
Public Shared Function FromBoolean( _
    val As Boolean _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter.FromByte Method

Converts the specified byte value to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromByte (
    byte val
)
```

Visual Basic Copy Code

```
Public Shared Function FromByte( _
    val As Byte _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromChar Method

Converts the specified char value to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromChar (
    char val
)
```

Visual Basic Copy Code

```
Public Shared Function FromChar( _
    val As Char _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromColor Method

Converts the specified color value to string in the format #AARRGGBB.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromColor (
    Color val
)
```

Visual Basic Copy Code

```
Public Shared Function FromColor( _
    val As Color _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromDecimal Method

Converts the specified decimal to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromDecimal (
    decimal val
)
```

Visual Basic Copy Code

```
Public Shared Function FromDecimal( _
    val As Decimal _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromDouble Method

Converts the specified double value to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromDouble (
    double val
)
```

Visual Basic Copy Code

```
Public Shared Function FromDouble( _
    val As Double _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromEnum Method

Converts the specified enum value to a string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromEnum (
    Object val
)
```

Visual Basic Copy Code

```
Public Shared Function FromEnum( _
    val As Object _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromImage Method

Converts the specified image to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromImage (
    IImage val
)
```

Visual Basic Copy Code

```
Public Shared Function FromImage( _
    val As IImage _
) As String
```

Parameters

val
The **Image** object to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromInt16 Method

Converts the specified 16-bit integer value to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromInt16 (
    short val
)
```

Visual Basic Copy Code

```
Public Shared Function FromInt16( _
    val As Short _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromInt32 Method

Converts the specified 32-bit integer value to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromInt32 (
    int val
)
```

Visual Basic Copy Code

```
Public Shared Function FromInt32( _
    val As Integer _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromInt64 Method

Converts the specified 64-bit integer to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromInt64 (
    long val
)
```

Visual Basic Copy Code

```
Public Shared Function FromInt64( _
    val As Long _
```

```
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromSByte Method

Converts the specified 8-bit signed integer to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromSByte (
    sbyte val
)
```

Visual Basic Copy Code

```
Public Shared Function FromSByte( _
    val As SByte _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromSingle Method

Converts the specified float value to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromSingle (
    float val
)
```

Visual Basic Copy Code

```
Public Shared Function FromSingle( _
    val As Single _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromStream Method

Converts the specified stream to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromStream (
    Stream val
)
```

Visual Basic Copy Code

```
Public Shared Function FromStream( _
    val As Stream _
) As String
```

Parameters

val
The **Stream** object to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromStreamV4 Method

Converts the specified memory stream to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromStreamV4 (
    MemoryStream val
)
```

Visual Basic

```
Public Shared Function FromStreamV4( _
    val As MemoryStream _
) As String
```

Parameters

val
The **Stream** object to convert.

Return Value

A string representing the specified value.

Remarks

This method is used in XML file format 4 and earlier.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromUInt16 Method

Converts the specified unsigned 16-bit integer value to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static string FromUInt16 (
    ushort val
)
```

Visual Basic

```
Public Shared Function FromUInt16( _
    val As UShort _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromUInt32 Method

Converts the specified unsigned 32-bit integer value to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromUInt32 (
    uint val
)
```

Visual Basic Copy Code

```
Public Shared Function FromUInt32( _
    val As UInteger _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.FromUInt64 Method

Converts the specified unsigned 64-bit integer value to string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static string FromUInt64 (
    ulong val
)
```

Visual Basic Copy Code

```
Public Shared Function FromUInt64( _
    val As ULong _
) As String
```

Parameters

val
The value to convert.

Return Value

A string representing the specified value.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ImageFromString Method

Converts the specified string value to image.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static IImage ImageFromString (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ImageFromString( _
    s As String _
) As IImage
```

Parameters

s
The string value to convert from.

Return Value

A **Microsoft.Maui.Graphics.IImage** that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToBoolean Method

Converts the specified string value to bool.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static bool ToBoolean (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToBoolean( _
    s As String _
) As Boolean
```

Parameters

s
The string value to convert from.

Return Value

A boolean value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToByte Method

Converts the specified string value to byte value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static byte ToByte (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToByte( _
    s As String _
) As Byte
```

Parameters

s
The string value to convert from.

Return Value

A byte value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToChar Method

Converts the specified string value to char.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static char ToChar (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToChar( _
    s As String _
) As Char
```

Parameters

s
The string value to convert from.

Return Value

A char value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToColor Method

Converts the specified string value to Color.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Color ToColor (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToColor( _
    s As String _
) As Color
```

Parameters

s
The string value to convert from.

Return Value

A **Color** value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToDecimal Method

Converts the specified string value to decimal.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static decimal ToDecimal (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToDecimal( _
    s As String _
) As Decimal
```

Parameters

s
The string value to convert from.

Return Value

A decimal value that is deserialized from the specified string.

See Also

Converter.ToDouble Method

Converts the specified string value to double.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double ToDouble (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToDouble( _
    s As String _
) As Double
```

Parameters

s
The string value to convert from.

Return Value

A double value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToEnum Method

Converts the specified string value to its corresponding enumeration member of the specified type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Object ToEnum (
    Type type,
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToEnum( _
    type As Type, _
    s As String _
) As Object
```

Parameters

type
The type of the enumeration to convert to.

s

The string value to convert from.

Return Value

A member of the specified enumeration that is deserialized from the specified string, possibly a **null** reference (**Nothing** in Visual Basic) if the deserialization failed.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToImage Method

Converts the specified string value to image.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static IImage ToImage (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToImage( _
    s As String _
) As IImage
```

Parameters

s
The string value to convert from.

Return Value

A **Microsoft.Maui.Graphics.IImage** that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToInt16 Method

Converts the specified string value to 16-bit integer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static short ToInt16 (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToInt16( _
    s As String _
) As Short
```

Parameters

s
The string value to convert from.

Return Value

A 16-bit integer value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToInt32 Method

Converts a string value to 32-bit integer.

Overload List

Name	Description
Converter.ToInt32 (String)	Converts the specified string value to 32-bit integer.
Converter.ToInt32 (String, Int32)	Converts the specified string value to 32-bit integer without throwing exceptions. In case the specified string cannot be converted, the specified default value is returned.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToInt32 Method (String)

Converts the specified string value to 32-bit integer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static int ToInt32 (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToInt32( _
    s As String _
) As Integer
```

Parameters

s
The string value to convert from.

Return Value

A 32-bit integer value that is deserialized from the specified string.

See Also

ToInt32 Method Overload List
Converter Members

Converter.ToInt32 Method (String, Int32)

Converts the specified string value to 32-bit integer without throwing exceptions. In case the specified string cannot be converted, the specified default value is returned.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static int ToInt32 (
    string s,
    int defaultValue
)
```

Visual Basic Copy Code

```
Public Shared Function ToInt32( _
    s As String, _
    defaultValue As Integer _
) As Integer
```

Parameters

s
The string value to convert from.

defaultValue
The default value to return if *s* cannot be deserialized.

Return Value

A 32-bit integer value that is deserialized from the specified string.

See Also

ToInt32 Method Overload List
Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToInt64 Method

Converts the specified string value to 64-bit integer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static long ToInt64 (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToInt64( _
    s As String _
) As Long
```

Parameters

s
The string value to convert from.

Return Value

A 64-bit integer value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToSByte Method

Converts the specified string value to 8-bit signed integer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static sbyte ToSByte (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToSByte( _
    s As String _
) As SByte
```

Parameters

s
The string value to convert from.

Return Value

An 8-bit signed integer value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToSingle Method

Converts the specified string value to float.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static float ToSingle (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToSingle( _
    s As String _
) As Single
```

Parameters

s
The string value to convert from.

Return Value

A float value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToStream Method

Converts the specified string value to stream.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Stream ToStream (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToStream( _
    s As String _
) As Stream
```

Parameters

s
The string value to convert from.

Return Value

A .NET **Stream** object that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToStreamV4 Method

Converts the specified string value to stream.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Stream ToStreamV4 (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToStreamV4( _
    s As String _
```

```
) As Stream
```

Parameters

s
The string value to convert from.

Return Value

A .NET **Stream** object that is deserialized from the specified string.

Remarks

This method is used in XML file format 4 and earlier.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToUint16 Method

Converts the specified string value to unsigned 16-bit integer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static ushort ToUint16 (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToUint16( _
    s As String _
) As UShort
```

Parameters

s
The string value to convert from.

Return Value

An unsigned 16-bit integer value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToUint32 Method

Converts the specified string value to unsigned 32-bit integer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static uint ToUint32 (
    string s
```

```
)
```

Visual Basic Copy Code

```
Public Shared Function ToUint32( _
    s As String _
) As UInteger
```

Parameters

s
The string value to convert from.

Return Value

An unsigned 32-bit integer value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

Converter.ToUint64 Method

Converts the specified string value to unsigned 64-bit integer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static ulong ToUint64 (
    string s
)
```

Visual Basic Copy Code

```
Public Shared Function ToUint64( _
    s As String _
) As ULong
```

Parameters

s
The string value to convert from.

Return Value

An unsigned 64-bit integer value that is deserialized from the specified string.

See Also

Converter Members
Converter Class
MindFusion.Diagramming Namespace

CreatePathDelegate Delegate

Specifies the signature of callback functions used to generate dynamic node shapes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public delegate IGraphicsPath CreatePathDelegate (
    Rect nodeBounds
)
```

Visual Basic Copy Code

```
Public Delegate Function CreatePathDelegate( _
    nodeBounds As Rect _
) As IGraphicsPath
```

Parameters

nodeBounds
A Rect object.

Return Value

A IGraphicsPath specifying the shape outline or decorations.

See Also

MindFusion.Diagramming Namespace

CustomDraw Enumeration

Specifies what kind of custom painting to perform on diagram elements.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum CustomDraw
```

Visual Basic Copy Code

```
Public Enum CustomDraw
```

Members

	Member name	Description
	None	Do not perform custom drawing, let all painting be done by MvcDiagram.
	Additional	The custom painting code executes after item's interior and frame are painted by MvcDiagram, but before text is rendered. In this mode, the **Graphics** instance passed to the custom drawing event handler has a clipping region applied to it.
	Full	The component does not do any standard drawing, the entire painting of an item is done by the application.
	ShadowOnly	Only the shadow of an item is painted by custom code, the item itself is drawn by the component.
	Additional2	The custom painting code executes after all parts of an item are painted by MvcDiagram. In this mode, the **Graphics** instance passed to the custom drawing event handler doesn't have a clipping region applied to it.

Remarks

These values can be assigned to the **CustomDraw** property of shapes, tables and links.

See Also

MindFusion.Diagramming Namespace

DelKeyAction Enumeration

Specifies the type of action performed when the user presses the DEL key.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum DelKeyAction
```

Visual Basic Copy Code

```
Public Enum DelKeyAction
```

Members

	Member name	Description
	None	Indicates that nothing happens.
	DeleteActiveItem	Indicates that only the ActiveItem is deleted.
	DeleteSelectedItems	Indicates that all selected items are deleted.

See Also

MindFusion.Diagramming Namespace

Diagram Class

Represents a flow diagram.

Syntax

C# Copy Code

```
public class Diagram : DiagramBase
```

Visual Basic Copy Code

```
Public Class Diagram
    Inherits DiagramBase
```

Remarks

An instance of the **Diagram** class represents a diagram document. A **Diagram** instance does not provide its own user interface. To display a diagram on a form, add a **DiagramView** instance to the form and set the view's **Diagram** property. A diagram can be displayed in more than one view.

A diagram can contain connected nodes and links. Nodes are instances of DiagramNode-derived classes. Nodes are stored in the Nodes collection. Links in MvcDiagram are instances of the DiagramLink class and are stored in the Links collection. Diagram items can be grouped in hierarchical structures and the so formed groups can be accessed by means of the Groups collection.

Managing Items

Items can be created by using the **new** operator and added to the diagram by means of the **Add** method of the Nodes or Links collection. Items are deleted by using the **Remove** method of the respective collection. There are shortcut methods provided by the diagram's Factory object that can be used to create an item and add it to the diagram with just one method call. Use the CreateTableNode, CreateShapeNode and CreateDiagramLink factory methods to create new items of the respective type.

Hierarchical groups of items are created by adding a Group instance to the diagram's Groups collection. The **Factory** object provides a shortcut CreateGroup method to do that with a single method call. Use the Remove method of the **Groups** collection to destroy a group.

Searching for a particular kind of items is done with the FindLink, FindNode and FindGroup methods, which look for an item that has a particular Tag, or FindLinkById and FindNodeById to find items by their Id. To get the item placed at a certain location, use

the GetItemAt method.

Serialization

Binary serialization of diagrams is provided by the SaveToStream and LoadFromStream methods which serialize diagrams to / from .NET streams. SaveToFile and LoadFromFile wrap their stream counterparts to implement easy serialization to disk files. SaveToString and LoadFromString store or load diagrams to/from ASCII strings, allowing easy storage of whole diagrams in a database or their transfer between a web-server and a browser through a hidden html element.

Use the SaveToXml and LoadFromXml methods to serialize a diagram into/from XML documents.

Controlling Users' Actions

Most aspects of user interaction are controlled by the **DiagramView** class.

Users actions can be validated by handling various events, fired while items are being created, modified or deleted.

The following properties control what kind of links users can create: AllowLinksRepeat, AllowSelfLoops, AllowUnanchoredLinks, AllowUnconnectedLinks.

The order in which items are hit-tested when clicking on them depends on the value of HitTestPriority. The LinkHitDistance property specifies the furthest distance from a link that a click would be still considered a hit.

Alignment

Items are aligned to a grid when AlignToGrid is enabled. To display the grid, set ShowGrid to true. The grid size and appearance can be customized via the GridSizeX, GridSizeY, GridColor and GridStyle properties.
If AutoAlignNodes is enabled, nodes are aligned to other nodes so that a pair of their vertical or horizontal sides become collinear. To specify the maximal distance at which nodes are aligned, set the AutoAlignDistance property. The pair of aligned nodes is indicated visually by drawing an alignment guide using the pen assigned to AlignmentGuidePen.

Links can also automatically snap to nodes when AutoSnapLinks is set. The maximal distance at which links will connect to a nearby node is set via AutoSnapDistance.

Appearance

The **Diagram** background area can be painted using a solid, gradient, texture or color brush, by setting the BackBrush property, and/or using an image specified by BackgroundImage and BackgroundImageAlign. Some more elaborate visual effects can be applied by means of the EffectsProvider, EnableEmbossEffects and EnableShadowEffects properties.

To further customize the diagram appearance, handle the DrawBackground event to render custom background graphics, or the DrawForeground event to render custom foreground graphics.

Lanes

The lane grid lets you emphasize the relationship between a group of diagram items by displaying them in a distinct lane or cell within the grid. Use the EnableLanes property to display a background grid of lanes, optionally divided into cells. The grid can be set up using the LaneGrid property.

Text Formatting

Use TextFormat and TextColor to specify alignment and color of the text for all diagram items. EnableStyledText allows using HTML-like tags to format item's text. PolygonalTextLayout allows laying text out inside non-rectangular node shapes.

Tweaks

- While moving grouped items, they are drawn in depth conforming to either their z-order or their group hierarchy, as set in SortGroupsByZ.
- SelectionOnTop specifies whether selected items should be painted on top of other items.
- MergeThreshold specifies how closely to each other adjacent link points should be placed in order to merge link segments.

Default Item Properties

Many properties of the **Diagram** class provide initial values for the properties of new items.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.DiagramBase
        MindFusion.Diagramming.Diagram
          MindFusion.Diagramming.DiagramPage
          MindFusion.Diagramming.DiagramPage

See Also

Diagram Members
MindFusion.Diagramming Namespace

Diagram Members

Represents a flow diagram.

The following tables list the members exposed by the Diagram type.

Public Constructors

	Name	Description
	Diagram	The Diagram class represents a flow diagram.

Public Fields

	Name	Description
	Dummy	A shared DummyNode instance.

Public Properties

	Name	Description
	ActiveItem	Gets or sets the active diagram element.
	ActiveItemHandlesStyle	Gets the visual style of adjustment handles of the active item.
	ActiveLayer	Gets or sets the zero-based index of the active layer.
	AdjustmentHandlesSize	Gets or sets the size of the handles, which appear around selected objects.
	AlignmentGuidePen	Gets or sets the pen used to draw alignment guides.
	AlignToGrid	Gets or sets a value indicating whether the diagram items is aligned to a grid.
	AllowLinksRepeat	Gets or sets a value indicating whether more than one link can connect the same origin and destination diagram nodes.
	AllowMultipleResize	Gets or sets whether multiple selected nodes can be resized simultaneously.
	AllowSelfLoops	Gets or sets a value indicating whether self-loops are allowed.
	AllowSplitLinks	Gets or sets a value indicating whether link segments can be added and removed interactively.
	AllowUnanchoredLinks	Gets or sets a value indicating whether links can be connected to nodes to which there aren't anchor points assigned.
	AllowUnconnectedLinks	Gets or sets a value indicating whether users are allowed to draw links that are not connected to any node.
	AutoAlignDistance	Gets or sets the maximal horizontal or vertical distance at which a node aligns to the borders of other nodes.
	AutoAlignNodes	Gets or sets whether nodes should automatically align to the borders of other nodes during user interaction.
	AutoArrangeAvoidContainers	Gets or sets a value indicating whether auto-arranged link labels should avoid containers.
	AutoArrangeAvoidSegments	Gets or sets a value indicating whether auto-arranged link labels should avoid link segments.
	AutoCloseDistance	Gets or sets the maximum distance between first and last points of a FreeFormNode for which the node's outline is closed automatically.
	AutoDeleteLinks	Gets or sets whether links are deleted automatically when their Origin or Destination is deleted.
	AutoHighlightRows	Gets or sets whether a table row should be highlighted when the user clicks on it.
	AutoResize	Gets or sets a value indicating whether the document area must be resized automatically so it fits the diagram contents.

	AutoSnapDistance	Specifies the maximum distance at which links auto-snap to nodes.
	AutoSnapLinks	Gets or sets the default value for the AutoSnapToNode property of new links.
	BackBrush	Gets or sets the brush used for painting the diagram background.
	BackgroundImage	Gets or sets the image displayed as background of this diagram.
	BackgroundImageAlign	Gets or sets a value indicating how the background image is positioned, relatively to the scrollable document area.
	BackgroundImageUrl	URL of the image painted as diagram's background.
	BarcodeNodeStyle	Gets the theme-level style for barcode nodes.
	Bounds	Gets or sets a value representing the size of the diagram's scrollable area.
	BoundsPen	The pen with which the document bounds should be drawn.
	BranchIndicatorColor	Gets or sets the color of link branch indicators.
	BranchIndicatorSize	Gets or sets the size of link branch indicators.
	CellCustomDraw	Gets or sets the default type of custom painting done on the cells of tables.
	CellFrameStyle	Gets or sets the default cell border style of new tables.
	CompositeNodeStyle	Gets the theme-level style for composite nodes.
	ContainerCaption	Gets or sets the default value for the Caption property of containers.
	ContainerCaptionHeight	Gets or sets the default value for the CaptionHeight property of containers.
	ContainerHighlightPen	Gets or sets the default value for the HighlightPen property of containers.
	ContainerMargin	Gets or sets the default value for the Margin property of containers.
	ContainerMinimumSize	Gets or sets the default value for the MinimumSize property of containers.
	ContainerNodeStyle	Gets the theme-level style for container nodes.
	ContainersFoldable	Gets or sets the default value for the Foldable property of containers.
	CrossingRadius	Gets or sets the radius length of decorations displayed at link intersection points.
	DefaultShape	Gets or sets the default shape of shape nodes.
	DiagramLinkStyle	Gets the theme-level style for diagram links.
	DiagramStyle	Gets the theme-level style for the diagram.
	Dirty	Gets or sets a value indicating whether the diagram has changed since the last save.
	DisabledHandlesStyle	Gets the visual style of disabled adjustment handles.
	DynamicLinks	Gets or sets the default value for the Dynamic property of new links.
	EffectiveBackBrush	Gets the currently effective background brush of the diagram.
	EffectsProvider	Gets or sets an object used to generate visual effects for shape nodes such as embossment and shadow dispersion.
	EnableEmbossEffects	Gets or sets a value indicating whether emboss effect for shape nodes is enabled.
	EnableLanes	Gets or sets whether lanes are enabled.

	EnableShadowEffects	Gets or sets a value indicating whether the shadow dispersion effect for nodes is enabled.
	EnableSpatialIndex	Gets or sets whether to index item positions for faster hit-testing and viewport clipping queries.
	EnableStyledText	Gets or sets the default value for the EnableStyledText property of new nodes.
	ExpandButtonAction	Gets or sets a value indicating what should happen when a user clicks the [±] button that is displayed near expandable nodes.
	ExpandButtonPosition	Gets or sets where expand/collapse buttons are located relatively to their nodes.
	ExpandOnIncoming	Gets or sets a value, indicating in which direction object hierarchies are expanded.
	ExteriorBrush	Gets or sets the brush used for painting the area outside **Bounds**.
	Factory	Gets a Factory instance that lets you add programmatically new items to the diagram.
	FocusedComponent	Gets or sets the component within this diagram which currently has the input focus.
	FoldedContainerSize	Gets or sets the size of folded containers.
	Font	Gets or sets the default font for new diagram items.
	FreeFormAttractDistance	Gets or sets the radius around dragged free-form adjustment handle in which other points of FreeFormNode are modified too.
	FreeFormTargets	An array of Shape objects used to replace FreeFormNode instances with ShapeNode ones when Behavior is set to LinkFreeShapes or DrawFreeShapes.
	GridColor	Gets or sets the color of the grid points.
	GridOffsetX	Gets or sets the horizontal offset of the first point of the alignment grid.
	GridOffsetY	Gets or sets the vertical offset of the first point of the alignment grid.
	GridPatternHatch	Gets or sets a hatch style used to draw dense alignment grid at low zoom levels.
	GridPatternThreshold	Gets or sets the distance between grid points (in pixels) that triggers drawing using **GridPatternHatch**.
	GridSizeX	Gets or sets the horizontal distance between adjacent grid points.
	GridSizeY	Gets or sets the vertical distance between adjacent grid points.
	GridStyle	Gets or sets the visual style of the alignment grid.
	Groups	Gets a collection of all groups in the diagram.
	HitTestPriority	Gets or sets a value indicating the order in which items are hit-tested.
	Interaction	Gets an InteractionState instance representing the current action being carried out by the user.
	IsProEdition	Gets a value indicating whether the currently loaded MvcDiagram is a professional version of the control.
	Items	Gets a collection of all items in this diagram.
	LaneGrid	Gets the Grid instance that represents the lane grid.
	Layers	Gets a list of the layers in the current diagram.
	LinkBaseShape	Gets or sets the shape displayed at the base of new links.
	LinkBaseShapeSize	Gets or sets the default size of the base of new links.

	LinkBranchIndicator	Gets or sets a value indicating what shapes to display as markers at points where overlapping Cascading links diverge.
	LinkBrush	Gets or sets the default value for the Brush property of links.
	LinkCascadeOrientation	Gets or sets the initial orientation of the first segment of new cascading links.
	LinkCrossings	Gets or sets how to render the intersection points where links cross their paths.
	LinkCustomDraw	Gets or sets the default type of custom painting done on links.
	LinkEndsMovable	Gets or sets a value indicating whether link ends can be moved by users after the link is created.
	LinkHandlesStyle	Gets or sets the default value for the HandlesStyle property of new links.
	LinkHeadShape	Gets or sets the shape displayed as head of new links.
	LinkHeadShapeSize	Gets or sets the default size of arrowheads.
	LinkHitDistance	Gets or sets a value specifying how far from a link a click is still considered a hit.
	LinkIntermediateShape	Gets or sets the default shape to be displayed in the middle of each segment of links.
	LinkIntermediateShapeSize	Gets or sets the default size of the shapes at middles of link segments.
	LinkMergeIndicator	Gets or sets a value indicating what shapes to display as markers at points where overlapping **Cascading** links merge.
	LinkPen	Gets or sets the default value for the Pen and HeadPen properties of links.
	LinkRouter	Gets or sets the component used to find paths for auto-routed links when diagram nodes are added or their positions change.
	Links	Gets a collection of all DiagramLink objects in the diagram.
	LinkSegments	Gets or sets the default number of link segments.
	LinkShape	Gets or sets the default value of the Shape property of new links.
	LinksRetainForm	Gets or sets whether links should maintain their forms if the nodes linked with those links are moved around.
	LinkText	Gets or sets the default text of links.
	LinkTextStyle	Gets or sets default placement and orientation for links' text.
	MeasureTextRenderingHint	Gets or sets the TextRenderingHint to use when measuring texts. This should be effectively set to the value of DiagramView.TextRenderingHint.
	MeasureUnit	Gets or sets the unit of measure used for logical coordinates in this diagram.
	MergeIndicatorColor	Gets or sets the color of link merge indicators.
	MergeIndicatorSize	Gets or sets the size of link merge indicators.
	MergeThreshold	Gets or sets the maximum distance between adjacent control points of a link at which the respective segments can be merged.
	MinimumNodeSize	Gets or sets the minimum node size that the user is allowed to set.
	ModificationEffect	Gets or sets the visual effect to apply during interactive modification of items.
	NodeEffects	Gets a collection with all effects applied to the nodes of this diagram.
	Nodes	Gets a collection of all DiagramNode objects in the diagram.

	NodesExpandable	Gets or sets a value indicating whether newly created nodes can be expanded.
	NowLoading	Gets a value indicating whether the diagram data is now being loaded from a file.
	PolygonalTextLayout	Gets or sets a value indicating whether to enable the formatting of text inside non-rectangular shape node outlines.
	RecursiveExpand	Gets or sets value indicating how a tree is expanded.
	RenderOptions	Gets the current rendering options for this diagram.
	RestrictItemsToBounds	Gets or sets a value indicating whether users are allowed to place items outside the scrollable area of the diagram.
	RoundedLinks	Gets or sets a value indicating whether the joins between segments of **Polyline** and **Cascading** links should be rounded.
	RoundedLinksRadius	Gets or sets the radius of arc joins of rounded links' segments.
	RoundRectFactor	Gets or sets the relative size of arcs displayed at the corners of rounded rectangles.
	RouteLinks	Gets or sets a value indicating whether links should avoid nodes by going the shortest path from their origin to their destination without crossing any other nodes.
	RoutingOptions	Gets an object, which allows customizing various parameters of the link routing algorithm.
	RowHighlightBrush	Gets or sets the Brush used to fill highlighted table rows.
	ScriptHelper	Gets a ScriptHelper object which can be used in html scripts to instantiate various objects and to call static methods.
	SelectAfterCreate	Gets or sets a value indicating whether newly created objects are automatically selected.
	SelectedItemHandlesStyle	Gets the visual style of adjustment handles of selected items.
	Selection	Gets a Selection instance that represents the selection of items in this diagram.
	SelectionOnTop	Gets or sets a value indicating whether selected items should be painted on top of other items.
	SerializationBinder	Gets or sets a SerializationBinder object used to control class loading.
	ShadowBrush	Gets or sets the default brush used to paint the shadows cast by diagram items.
	ShadowOffsetX	Gets or sets the horizontal offset of item shadows.
	ShadowOffsetY	Gets or sets the vertical offset of item shadows.
	ShadowsStyle	Gets or sets the style for rendering shadows.
	ShapeBrush	Gets or sets the default value for the **Brush** property of shape nodes.
	ShapeCustomDraw	Gets or sets the default type of custom painting done for shape nodes.
	ShapeHandlesStyle	Gets or sets the default value for the **HandlesStyle** property of new shape nodes.
	ShapeNodeStyle	Gets the theme-level style for shape nodes.
	ShapeOrientation	Gets or sets the default value for the ShapeOrientation property of shape nodes.
	ShapePen	Gets or sets the default value for the **Pen** property of shape nodes.
	ShapeText	Gets or sets the default text of shape nodes.

	ShowAnchors	Gets or sets a value indicating whether and when node anchor points should be displayed.
	ShowDisabledHandles	Gets or sets a value indicating whether manipulation handles are painted when they are disabled.
	ShowGrid	Gets or sets a value indicating whether the alignment grid is visible.
	ShowHandlesOnDrag	Gets or sets a value indicating whether to display selection handles of objects under the mouse while another object is being modified.
	SnapToAnchor	Gets or sets when link end points should be aligned to the nearest anchor points.
	SortGroupsByZ	Gets or sets a value indicating whether grouped items should be drawn sorted by their z-order while the whole group is being moved.
	Style	Gets or sets the style associated with the diagram.
	SvgNodeStyle	Gets the theme-level style for SVG nodes.
	TableBrush	Gets or sets the default value for the **Brush** property of tables.
	TableCaption	Gets or sets the default text for tables' captions.
	TableCaptionHeight	Gets or sets the default height of table captions.
	TableColumnCount	Gets or sets the default number of table columns.
	TableColumnWidth	Gets or sets the default width of tables' columns.
	TableConnectionStyle	Gets or sets a value indicating whether drawing a link between tables should connect the rows of those tables, or the tables as whole entities.
	TableCustomDraw	Gets or sets the default type of custom painting done on tables.
	TableHandlesStyle	Gets or sets the default value for the **HandlesStyle** property of new tables.
	TableNodeStyle	Gets the theme-level style for table nodes.
	TablePen	Gets or sets the default value for the **Pen** property of tables.
	TableRowCount	Gets or sets the default number of table rows.
	TableRowHeight	Gets or sets the default height of tables' rows.
	TableShape	Gets or sets the default value of the Shape property of new tables.
	TablesScrollable	Gets or sets the default value for the Scrollable property of new tables.
	Tag	Gets or sets custom data associated with this diagram.
	TextColor	Gets or sets the default color of text displayed inside diagram items.
	TextFormat	Gets or sets the default alignment and formatting style for text displayed inside nodes and table cells.
	Theme	Gets or sets the theme associated with the diagram.
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram.
	TouchHitDistance	Specifies distance from adjustment handles at which they are still considered as hit when processing touch input.
	TouchThreshold	Gets or sets the threshold value for easier touch interaction through adjustment handles.
	TreeViewNodeStyle	Gets the theme-level style for tree view nodes.
	UndoManager	Gets a reference to the diagram's UndoManager instance.

	ValidityChecks	Gets or sets a value indicating whether validity checks are performed each time an item is added to the diagram.
	VisioNodeStyle	Gets the theme-level style for Visio nodes.
	XmlOptions	An XmlOptions object providing properties for customizing the XML serialization process. (Inherited from DiagramBase.)

Public Methods

	Name	Description
	AlignPointToGrid	Returns a point of the alignment grid nearest to the one passed as an argument.
	AllowNodeBounds	Determines whether node can be placed at specified coordinates.
	Arrange	Arranges the diagram using the specified graph layout algorithm.
	ArrangeAnimated	Arranges the diagram using the specified graph layout algorithm, and animates items moving to their new positions.
	ArrangeAsync	Arranges the diagram asynchronously using the specified graph layout algorithm.
	ArrangeLinkLabels	Rearranges link labels whose AutoArrange property is enabled.
	ChangeUnit	Sets new **MeasureUnit** value and scales the coordinates of diagram items so that their absolute size remains unchanged.
	ClearAll	Removes all items from the diagram.
	Clone	Creates a copy of this diagram.
	CopySelection	Overloaded. Creates a copy of specified diagram elements.
	CopyTo	Copies this diagram's properties and items to specified target diagram.
	CreateImage	Overloaded. Creates a bitmap containing an image of the diagram.
	CreateMeasureGraphics	Creates an IGraphics object that can be used to obtain various string size measurements.
	CreateThemeFromDefaults	Creates a Theme object representing the default properties of the diagram.
	Draw	Renders the diagram on the specified **IGraphics** object.
	DrawAlignmentGuides	Draws the current alignment guides.
	DrawInteraction	Draws the items that the user is currently modifying.
	DrawInteractions	For internal use.
	DrawLayer	Renders all diagram items to the specified graphics.
S	DrawStyledText	Draws a formatted text string, the same way text is drawn in shape nodes whose **EnableStyledText** property is set to **true**. This method lets you display styled text in custom-drawn nodes or links.
	EffectiveTouchHitDistance	Returns **TouchHitDistance** or a default value.
	EnumByReverseZ	Hit-tests diagram items in reverse Z order.
	EnumTopLevelByReverseZ	Hit-tests diagram top-level items in reverse Z order.
	ExecuteCommand	Executes the specified command on this diagram.
	FindGroup	Finds the group that has the specified tag value.
	FindLink	Finds the link that has the specified tag value.
	FindLinkById	Finds the link that has the specified id value.

	FindNode	Finds the node that has the specified tag value.
	FindNodeById	Finds the node that has the specified id value.
	GetContentBounds	Returns the smallest rectangle that bounds all diagram objects.
	GetItemAt	Overloaded.
	GetLinkAt	Overloaded. Finds and returns the link that lies at the specified location.
	GetLinksAt	Returns the links that contain the specified point.
	GetNearestNode	Returns the node nearest to the specified point.
	GetNodeAt	Overloaded. Gets the node that lies at the specified location.
	GetNodesAt	Returns the nodes that contain the specified point.
	GetNodesInViewport	Returns all nodes visible in the current viewport.
	HandleKeyDown	Propagates keyboard input to interested manipulators and components.
	HandleKeyPress	Propagates keyboard input to interested manipulators.
	HandleKeyUp	Propagates keyboard input to interested components.
	HandleMouseMove	Propagates mouse input to interested manipulators.
	HitTestComponents	Tests whether the specified point is within a component of a diagram item and returns this component.
	HitTestLabels	Returns the item label located at specified position.
	HitTestManipulators	Tests whether the specified point is within the manipulator of a diagram object and triggers the manipulator.
	Invalidate	Overloaded. Invalidates the diagram or a region of the diagram, causing it to be repainted.
	InvalidateZOrder	Forces a new sort of the Z order list before next redraw.
	IsClassRegistered	Checks if a class is registered for serialization support.
	IsClientClassRegistered	Checks if a class is registered for JSON serialization support.
	IsItemInteractive	Checks if the specified item can be modified interactively.
	IsItemLocked	Checks if the specified item is locked.
	IsItemVisible	Checks if the specified item is currently visible.
	LoadFromJson	Loads the diagram from a JSON-formatted string.
	LoadFromJsonAsync	Loads the diagram from a JSON-formatted string.
	LoadFromJsonFile	Loads the diagram from a JSON file.
	LoadFromJsonFileAsync	Loads the diagram from a JSON file.
	LoadFromJsonObject	Loads the diagram from a JSON object.
	LoadFromString	Decodes diagram contents from a string that was created with SaveToString.
	LoadFromXml	Overloaded. Loads the contents of an XML diagram document into the control.
	LoadFromXmlAsync	Overloaded.
	LoadSelectionFromJson	
	MeasureString	Overloaded. Measures a text taking into consideration the current affine transform of the diagram.

	Navigate	Selects the nearest item in the specified direction.
	NewInstance	Custom Diagram classes must override this method to return an instance of their type.
	PasteSelection	Adds the specified items and groups to the specified diagram document.
	RaiseCellTextEdited	Raises the CellTextEdited event .
	RaiseCellTextEditing	Raises the CellTextEditing event to validate entering in-place editing mode.
	RaiseClicked	Overloaded. Raises the Clicked event of the diagram or one of the *Clicked events related to diagram items.
	RaiseCreated	Raises the appropriate *Created event according to the type of the specified item.
	RaiseDeleting	Raises the appropriate *Deleting event according to the type of the specified item.
	RaiseDoubleClicked	Overloaded. Raises the DoubleClicked event of the diagram or one of the *DoubleClicked events related to diagram items.
	RaiseLinkTextEdited	Raises the LinkTextEdited event.
	RaiseLinkTextEditing	Raises the LinkTextEditing event to validate entering inplace edit mode.
	RaiseModified	Raises the NodeModified event.
	RaiseMouseHover	Raises the appropriate Pointed event.
	RaiseNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	RaiseNodeTextEdited	Raises the NodeTextEdited event.
	RaiseNodeTextEditing	Raises the NodeTextEditing event to validate entering inplace edit mode.
	RaisePasted	Raises the appropriate Pasted event.
	RaiseSelecting	Raises the appropriate Selecting event.
	RaiseTreeItemTextEditing	Raises the TreeItemTextEditing event to validate entering inplace edit mode.
	RaiseViewRemoved	Raises the ViewRemoved event.
	RegisterClass	Overloaded.
	RegisterDefaultStyle	Registers a default style for the objects of the specified type.
	RegisterItemClass	Overloaded.
	ResetAlignmentGuides	For internal use.
	ResizeToFitItem	Resizes the document extents so that the specified item is contained within.
	ResizeToFitItems	Overloaded. Resizes the document scrollable area so it fits all diagram items.
	ResumeLinkRouter	Called when the control ends a batch update operation that might affect many links.
	RouteAllLinks	Overloaded.
	SaveSelectionToJson	Saves the diagram selection to a JSON object.
	SaveToJson	Saves the diagram to a JSON-formatted string.
	SaveToJsonAsync	Saves the diagram to a JSON-formatted string.
	SaveToJsonFile	Saves the diagram into a JSON file.

	SaveToJsonFileAsync	Saves the diagram into a JSON file.
	SaveToJsonObject	Saves the diagram to a JSON object.
	SaveToString	Encodes the diagram contents into an ASCII string using the specified format and optionally clearing the **Dirty** flag.
	SaveToXml	Overloaded. Saves the diagram to an XML document.
	SaveToXmlAsync	Overloaded.
	SetShadowOffset	Sets the default offset at which items shadows are drawn.
	SuspendLinkRouter	Called when the control starts a batch update operation that might affect many links. The diagram collects links specified through Route and RouteAllLinks calls and defers finding their paths to when **ResumeLinkRouter** is called.
	TrackMouseHover	Specifies whether the diagram view should track mouse hover events. (Inherited from DiagramBase.)
	UpdateMeasureTextScale	Gets or sets the scale to use when measuring texts. This should be effectively set to the value of DiagramView.Zoom / 100.
	UpdateRuntimeIndices	For internal use.
	UpdateZOrder	Immediately sorts the Z-order list if Z-order has been invalidated. Otherwise it will be sorted next time the diagram draws its contents.

Protected Methods

	Name	Description
	AddItem	Invoked by the AddItemCmd and RemoveItemCmd classes to effectively add the specified item to the diagram.
	CreateAddItemCmd	Creates an **AddItemCmd** instance that implements undo/redo of adding an item to the diagram.
	CreateModifyItemCmd	Creates a ModifyItemCmd instance that implements undo/redo of item move and resize operations.
	CreateRemoveItemCmd	Creates a **RemoveItemCmd** instance that implements undo/redo of removing an item from the diagram.
	DeleteGroup	Removes the specified group from the group collection.
	Dispose	Clean up any resources being used.
	GetValue	Gets the effective value of the style property with the specified name.
	OnActionRecorded	Raises the ActionRecorded event. (Inherited from DiagramBase.)
	OnActionRecording	Raises the ActionRecording event. (Inherited from DiagramBase.)
	OnActionRedone	Raises the ActionRedone event. (Inherited from DiagramBase.)
	OnActionUndone	Raises the ActionUndone event. (Inherited from DiagramBase.)
	OnBeginLoad	Raises the BeginLoad event. (Inherited from DiagramBase.)
	OnBoundsChanged	Raises the BoundsChanged event. (Inherited from DiagramBase.)
	OnCellClicked	Raises the CellClicked event. (Inherited from DiagramBase.)
	OnCellDoubleClicked	Raises the CellDoubleClicked event. (Inherited from DiagramBase.)
	OnCellTextEdited	Raises the **CellTextEdited** event. (Inherited from DiagramBase.)
	OnCellTextEditing	Raises the **CellTextEditing** event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnClicked	Raises the **Clicked** event. (Inherited from DiagramBase.)

	OnContainerChildAdded	Raises the ContainerChildAdded event. (Inherited from DiagramBase.)
	OnContainerChildRemoved	Raises the ContainerChildRemoved event. (Inherited from DiagramBase.)
	OnContainerFolded	Raises the ContainerFolded event. (Inherited from DiagramBase.)
	OnContainerPageChanged	Raises the ContainerPageChanged event. (Inherited from DiagramBase.)
	OnContainerUnfolded	Raises the ContainerUnfolded event. (Inherited from DiagramBase.)
	OnContentVisibilityChanged	Raises the ContentVisibilityChanged event. (Inherited from DiagramBase.)
	OnDefaultShapeChanged	Raises the DefaultShapeChanged event. (Inherited from DiagramBase.)
	OnDeserializeTag	Raises the DeserializeTag event. (Inherited from DiagramBase.)
	OnDestinationAnchorChanged	Raises the DestinationAnchorChanged event. (Inherited from DiagramBase.)
	OnDirtyChanged	Raises the DirtyChanged event. (Inherited from DiagramBase.)
	OnDoubleClicked	Raises the **DoubleClicked** event. (Inherited from DiagramBase.)
	OnDrawAdjustmentHandles	Raises the DrawAdjustmentHandles event. (Inherited from DiagramBase.)
	OnDrawAnchorPoint	Raises the DrawAnchorPoint event. (Inherited from DiagramBase.)
	OnDrawBackground	Raises the DrawBackground event. (Inherited from DiagramBase.)
	OnDrawBranchIndicator	Raises the DrawBranchIndicator event. (Inherited from DiagramBase.)
	OnDrawCell	Raises the DrawCell event. (Inherited from DiagramBase.)
	OnDrawForeground	Raises the DrawForeground event. (Inherited from DiagramBase.)
	OnDrawLink	Raises the DrawLink event. (Inherited from DiagramBase.)
	OnDrawNode	Raises the DrawNode event. (Inherited from DiagramBase.)
	OnEndLoad	Raises the EndLoad event. (Inherited from DiagramBase.)
	OnExpandButtonClicked	Raises the ExpandButtonClicked event. (Inherited from DiagramBase.)
	OnGroupDestroyed	Raises the GroupDestroyed event. (Inherited from DiagramBase.)
	OnHitTestAdjustmentHandles	Raises the HitTestAdjustmentHandles event. (Inherited from DiagramBase.)
	OnInitializeLasso	Raises the InitializeLasso event. (Inherited from DiagramBase.)
	OnInitializeLink	Raises the InitializeLink event. (Inherited from DiagramBase.)
	OnInitializeNode	Raises the InitializeNode event. (Inherited from DiagramBase.)
	OnItemAdded	Raises the ItemAdded event. (Inherited from DiagramBase.)
	OnItemRemoved	Raises the ItemRemoved event. (Inherited from DiagramBase.)
	OnJsonDeserializeTag	Raises the JsonDeserializeTag event. (Inherited from DiagramBase.)
	OnJsonSerializeTag	Raises the JsonSerializeTag event. (Inherited from DiagramBase.)
	OnLayerVisibilityChanged	Raises the LayerVisibilityChanged event. (Inherited from DiagramBase.)
	OnLinkActivated	Raises the LinkActivated event. (Inherited from DiagramBase.)
	OnLinkClicked	Raises the LinkClicked event. (Inherited from DiagramBase.)
	OnLinkCreateCancelled	Raises the LinkCreateCancelled event. (Inherited from DiagramBase.)
	OnLinkCreated	Raises the LinkCreated event. (Inherited from DiagramBase.)
	OnLinkCreating	Raises the LinkCreating event. (Inherited from DiagramBase.)

	OnLinkDeactivated	Raises the LinkDeactivated event. (Inherited from DiagramBase.)
	OnLinkDeleted	Raises the LinkDeleted event. (Inherited from DiagramBase.)
	OnLinkDeleting	Raises the LinkDeleting event. (Inherited from DiagramBase.)
	OnLinkDeselected	Raises the LinkDeselected event. (Inherited from DiagramBase.)
	OnLinkDoubleClicked	Raises the LinkDoubleClicked event. (Inherited from DiagramBase.)
	OnLinkHovered	Raises the LinkHovered event. (Inherited from DiagramBase.)
	OnLinkModified	Raises the LinkModified event. (Inherited from DiagramBase.)
	OnLinkModifyCancelled	Raises the LinkModifyCancelled event. (Inherited from DiagramBase.)
	OnLinkModifying	Raises the LinkModifying event. (Inherited from DiagramBase.)
	OnLinkPasted	Raises the LinkPasted event. (Inherited from DiagramBase.)
	OnLinkRouted	Raises the LinkRouted event. (Inherited from DiagramBase.)
	OnLinkSelected	Raises the LinkSelected event. (Inherited from DiagramBase.)
	OnLinkSelecting	Raises the LinkSelecting event. (Inherited from DiagramBase.)
	OnLinkSplit	Raises the LinkSplit event. (Inherited from DiagramBase.)
	OnLinkStartModifying	Raises the LinkStartModifying event. (Inherited from DiagramBase.)
	OnLinkTextEdited	Raises the **LinkTextEdited** event. (Inherited from DiagramBase.)
	OnLinkTextEditing	Raises the **LinkTextEditing** event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnMeasureUnitChanged	Raises the MeasureUnitChanged event. (Inherited from DiagramBase.)
	OnNodeActivated	Raises the NodeActivated event. (Inherited from DiagramBase.)
	OnNodeClicked	Raises the NodeClicked event. (Inherited from DiagramBase.)
	OnNodeCreateCancelled	Raises the NodeCreateCancelled event. (Inherited from DiagramBase.)
	OnNodeCreated	Raises the NodeCreated event. (Inherited from DiagramBase.)
	OnNodeCreating	Raises the NodeCreating event. (Inherited from DiagramBase.)
	OnNodeDeactivated	Raises the NodeDeactivated event. (Inherited from DiagramBase.)
	OnNodeDeleted	Raises the NodeDeleted event. (Inherited from DiagramBase.)
	OnNodeDeleting	Raises the NodeDeleting event. (Inherited from DiagramBase.)
	OnNodeDeselected	Raises the NodeDeselected event. (Inherited from DiagramBase.)
	OnNodeDoubleClicked	Raises the NodeDoubleClicked event. (Inherited from DiagramBase.)
	OnNodeHovered	Raises the NodeHovered event. (Inherited from DiagramBase.)
	OnNodeModified	Raises the **NodeModified** event. (Inherited from DiagramBase.)
	OnNodeModifyCancelled	Raises the NodeModifyCancelled event. (Inherited from DiagramBase.)
	OnNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	OnNodePasted	Raises the NodePasted event. (Inherited from DiagramBase.)
	OnNodeSelected	Raises the NodeSelected event. (Inherited from DiagramBase.)
	OnNodeSelecting	Raises the NodeSelecting event. (Inherited from DiagramBase.)
	OnNodeStartModifying	Raises the NodeStartModifying event. (Inherited from DiagramBase.)

	OnNodeTextEdited	Raises the **NodeTextEdited** event. (Inherited from DiagramBase.)
	OnNodeTextEditing	Raises the NodeTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnOriginAnchorChanged	Raises the OriginAnchorChanged event. (Inherited from DiagramBase.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from DiagramBase.)
	OnQueryAlignTarget	Raises the QueryAlignTarget event. (Inherited from DiagramBase.)
	OnRepaint	Raises the Repaint event. (Inherited from DiagramBase.)
	OnRequestClipboard	Raises the RequestClipboard event. (Inherited from DiagramBase.)
	OnRequestDrawInteractions	Raises the RequestDrawInteractions event. (Inherited from DiagramBase.)
	OnRequestTimer	Raises the RequestTimer event. (Inherited from DiagramBase.)
	OnSelectionChanged	Raises the SelectionChanged event. (Inherited from DiagramBase.)
	OnSelectionModified	Raises the SelectionModified event. (Inherited from DiagramBase.)
	OnSelectionModifying	Raises the SelectionModifying event. (Inherited from DiagramBase.)
	OnSerializeTag	Raises the SerializeTag event. (Inherited from DiagramBase.)
	OnSetSelfLoopShape	Raises the SetSelfLoopShape event. (Inherited from DiagramBase.)
	OnShapeModified	Raises the ShapeModified event. (Inherited from DiagramBase.)
	OnShapeNotDefined	Raises the ShapeNotDefined event. (Inherited from DiagramBase.)
	OnTableColumnResized	Raises the TableColumnResized event. (Inherited from DiagramBase.)
	OnTableColumnResizing	Raises the TableColumnResizing event. (Inherited from DiagramBase.)
	OnTableRowResized	Raises the TableRowResized event. (Inherited from DiagramBase.)
	OnTableRowResizing	Raises the TableRowResizing event. (Inherited from DiagramBase.)
	OnTableSectionCollapsed	Raises the TableSectionCollapsed event. (Inherited from DiagramBase.)
	OnTableSectionExpanded	Raises the TableSectionExpanded event. (Inherited from DiagramBase.)
	OnTreeCollapsed	Raises the TreeCollapsed event. (Inherited from DiagramBase.)
	OnTreeExpanded	Raises the TreeExpanded event. (Inherited from DiagramBase.)
	OnTreeItemTextEditing	Raises the **TreeItemTextEditing** event. (Inherited from DiagramBase.)
	OnValidateAnchorPoint	Raises the ValidateAnchorPoint event. (Inherited from DiagramBase.)
	OnValidateLinkCrossing	Raises the ValidateLinkCrossing event. (Inherited from DiagramBase.)
	OnViewRemoved	Raises the **ViewRemoved** event. (Inherited from DiagramBase.)
	RemoveItem	Invoked by the **AddItemCmd** and **RemoveItemCmd** classes to effectively remove the specified item from the diagram.

Public Events

	Name	Description
	ActionRecorded	Raised when an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRecording	A validation event raised before an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRedone	Raised when an action is redone. (Inherited from DiagramBase.)

	ActionUndone	Raised when an action is undone. (Inherited from DiagramBase.)
	ActiveLayerChanged	Raised when the diagram's active layer has changed. (Inherited from DiagramBase.)
	BeginLoad	Raised when the component starts loading diagram data from a file or memory stream. (Inherited from DiagramBase.)
	BoundsChanged	Raised when the **Bounds** property value has changed. (Inherited from DiagramBase.)
	CellClicked	Raised when the user clicks a table's cell. (Inherited from DiagramBase.)
	CellDoubleClicked	Raised when the user double-clicks a table's cell. (Inherited from DiagramBase.)
	CellTextEdited	Raised when the user edits the text of a table's cell. (Inherited from DiagramBase.)
	CellTextEditing	A validation event that lets you approve or reject changes made to the text of a table's cell. (Inherited from DiagramBase.)
	Clicked	Raised when the user clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	ContainerChildAdded	Raised when a node is added to a ContainerNode, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerChildRemoved	Raised when a node is removed from a **ContainerNode**, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerFolded	Raised when the user folds a container by clicking the arrow-up button in the container's caption bar. (Inherited from DiagramBase.)
	ContainerPageChanged	Raised when the user changes current page of a PagedContainerNode. (Inherited from DiagramBase.)
	ContainerUnfolded	Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar. (Inherited from DiagramBase.)
	ContentVisibilityChanged	Raised when the visibility of container's content changes. (Inherited from DiagramBase.)
	DefaultShapeChanged	Raised when the **DefaultShape** property value has changed. (Inherited from DiagramBase.)
	DeserializeTag	Raised when the Tag of a diagram item must be deserialized from XML format. (Inherited from DiagramBase.)
	DestinationAnchorChanged	Raised when the DestinationAnchor property value of a link has changed. (Inherited from DiagramBase.)
	DirtyChanged	Raised when the **Dirty** property value has changed. (Inherited from DiagramBase.)
	DoubleClicked	Raised when the user double-clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	DrawAdjustmentHandles	Raised when item's adjustment handles must be custom-drawn. (Inherited from DiagramBase.)
	DrawAnchorPoint	Raised when an anchor point must be custom-drawn. (Inherited from DiagramBase.)
	DrawBackground	Raised to let you custom draw the diagram background. (Inherited from DiagramBase.)
	DrawBranchIndicator	Raised when a branch indicator must be custom-drawn. (Inherited from DiagramBase.)
	DrawCell	Raised when a table's cell must be custom drawn. (Inherited from DiagramBase.)

	DrawForeground	Raised to let you custom draw the diagram foreground. (Inherited from DiagramBase.)
	DrawLink	Raised when a diagram link must be custom drawn. (Inherited from DiagramBase.)
	DrawNode	Raised when a diagram node must be custom drawn. (Inherited from DiagramBase.)
	EndLoad	Raised when the component ends loading diagram data from a file or memory stream. (Inherited from DiagramBase.)
	ExpandButtonClicked	Raised when the user clicks the [±] button of an Expandable node. (Inherited from DiagramBase.)
	GroupDestroyed	Raised when a Group object is destroyed. (Inherited from DiagramBase.)
	HitTestAdjustmentHandles	Raised when custom hit testing of item's adjustment handles must be performed. (Inherited from DiagramBase.)
	InitializeLasso	Raised when the user starts drawing a selection lasso, before it is rendered on screen. (Inherited from DiagramBase.)
	InitializeLink	Raised when the user starts drawing a new link, just after the **DiagramLink** instance is created. (Inherited from DiagramBase.)
	InitializeNode	Raised when the user starts drawing a new node, just after the **DiagramNode** instance is created. (Inherited from DiagramBase.)
	ItemAdded	Raised when an item is added to the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	ItemRemoved	Raised when an item is removed from the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	JsonDeserializeTag	Raised when the **Tag** of a diagram item must be deserialized from JSON. (Inherited from DiagramBase.)
	JsonSerializeTag	Raised when the **Tag** of a diagram item must be serialized to JSON. (Inherited from DiagramBase.)
	LayerVisibilityChanged	Raised when the visibility of a Layer changes. (Inherited from DiagramBase.)
	LinkActivated	Raised when a link becomes the **ActiveItem**. (Inherited from DiagramBase.)
	LinkClicked	Raised when the user clicks a link. (Inherited from DiagramBase.)
	LinkCreateCancelled	Raised when the user cancels link creation. (Inherited from DiagramBase.)
	LinkCreated	Raised when the user draws a new link. (Inherited from DiagramBase.)
	LinkCreating	A validation event raised while the user is drawing a new link. (Inherited from DiagramBase.)
	LinkDeactivated	Raised when a link is no longer the **ActiveItem**. (Inherited from DiagramBase.)
	LinkDeleted	Raised when a link is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDeleting	Raised when the user tries to delete a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkDeselected	Raised when a link is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDoubleClicked	Raised when the user double-clicks a link. (Inherited from DiagramBase.)
	LinkHovered	Raised when the mouse pointer hovers over a link. (Inherited from DiagramBase.)

	LinkModified	Raised when the user moves any control point of a link. (Inherited from DiagramBase.)
	LinkModifyCancelled	Raised when the user cancels move or resize operation being done on a link. (Inherited from DiagramBase.)
	LinkModifying	A validation event raised while the user is modifying a link. (Inherited from DiagramBase.)
	LinkPasted	Raised when a link has been pasted from the clipboard. (Inherited from DiagramBase.)
	LinkRouted	Raised when a link is routed by the auto-routing algorithm. (Inherited from DiagramBase.)
	LinkSelected	Raised when a link is selected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkSelecting	Raised when the user tries to select a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkSplit	Raised when the user splits a link interactively. (Inherited from DiagramBase.)
	LinkStartModifying	A validation event raised when the user starts modifying a link. (Inherited from DiagramBase.)
	LinkTextEdited	Raised when the user edits the text of a link. (Inherited from DiagramBase.)
	LinkTextEditing	A validation event that lets you approve or reject changes made to the text of a link. (Inherited from DiagramBase.)
	MeasureUnitChanged	Raised when the **MeasureUnit** property value has changed. (Inherited from DiagramBase.)
	NodeActivated	Raised when a node becomes the **ActiveItem**. (Inherited from DiagramBase.)
	NodeClicked	Raised when the user clicks a node. (Inherited from DiagramBase.)
	NodeCreateCancelled	Raised when the user cancels node creation. (Inherited from DiagramBase.)
	NodeCreated	Raised when the user draws a new node. (Inherited from DiagramBase.)
	NodeCreating	A validation event raised while the user is drawing a new node. (Inherited from DiagramBase.)
	NodeDeactivated	Raised when a node is no longer the **ActiveItem**. (Inherited from DiagramBase.)
	NodeDeleted	Raised when a node is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDeleting	Raised when the user tries to delete a node, this event lets you cancel the operation. (Inherited from DiagramBase.)
	NodeDeselected	Raised when a node is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDoubleClicked	Raised when the user double-clicks a node. (Inherited from DiagramBase.)
	NodeHovered	Raised when the mouse pointer hovers over a node. (Inherited from DiagramBase.)
	NodeModified	Raised when the user moves or resizes a node. (Inherited from DiagramBase.)
	NodeModifyCancelled	Raised when the user cancels move or resize operation being done on a node. (Inherited from DiagramBase.)
	NodeModifying	A validation event raised while the user is moving or resizing a node. (Inherited from DiagramBase.)

	NodePasted	Raised when a node has been pasted from the clipboard. (Inherited from DiagramBase.)
	NodeSelected	Raised when a node is selected, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeSelecting	Raised when the user tries to select a node, this event lets you cancel the operation. (Inherited from DiagramBase.)
	NodeStartModifying	A validation event raised when the user starts modifying a node. (Inherited from DiagramBase.)
	NodeTextEdited	Raised when the user edits the text of a node. (Inherited from DiagramBase.)
	NodeTextEditing	A validation event that lets you approve or reject changes made to the text of a node. (Inherited from DiagramBase.)
	OriginAnchorChanged	Raised when the OriginAnchor property value of a link has changed. (Inherited from DiagramBase.)
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation. (Inherited from DiagramBase.)
	QueryAlignTarget	Raised to validate a node as target for alignment guides when **AutoAlignNodes** is enabled. (Inherited from DiagramBase.)
	Repaint	Raised when the diagram views should be repainted. (Inherited from DiagramBase.)
	RequestCacheReset	Raised when an object within the diagram requests a reset of the cached bitmap. (Inherited from DiagramBase.)
	RequestClipboard	Raised when an object within the diagram requests an operation with the clipboard. (Inherited from DiagramBase.)
	RequestDrawInteractions	For internal use. (Inherited from DiagramBase.)
	RequestTimer	Raised when an object within the diagram needs to start or stop a timer. (Inherited from DiagramBase.)
	SelectionChanged	Raised when an item is added to or removed from the selection (Inherited from DiagramBase.)
	SelectionModified	Raised when the user moves multiple selected items. (Inherited from DiagramBase.)
	SelectionModifying	Raised while the user is moving multiple selected items, this event lets you cancel the modification. (Inherited from DiagramBase.)
	SerializeTag	Raised when the **Tag** of a diagram item must be serialized into an XML document. (Inherited from DiagramBase.)
	SetSelfLoopShape	Raised when a link becomes a self-loop, giving you a chance to override the default look shape. (Inherited from DiagramBase.)
	ShapeModified	Raised when the user modifies the shape of a ShapeNode by moving one of the control points corresponding to shape formula parameters. (Inherited from DiagramBase.)
	ShapeNotDefined	Raised when the diagram being loaded contains nodes with an undefined shape. (Inherited from DiagramBase.)
	TableColumnResized	Raised when the user finishes resizing a column in a table. (Inherited from DiagramBase.)
	TableColumnResizing	Raised when the user resizes a column in a table interactively. (Inherited from DiagramBase.)
	TableRowResized	Raised when the user finishes resizing a row in a table. (Inherited from DiagramBase.)
	TableRowResizing	Raised when the user resizes a row in a table interactively. (Inherited from DiagramBase.)

	TableSectionCollapsed	Raised when the user collapses the rows of a table's section by clicking the [-] button in the section's header row. (Inherited from DiagramBase.)
	TableSectionExpanded	Raised when the user expands the rows of a table's section by clicking the [+] button in the section's header row. (Inherited from DiagramBase.)
	TreeCollapsed	Raised when a user collapses a tree branch by clicking the [-] button of an **Expandable** node. (Inherited from DiagramBase.)
	TreeExpanded	Raised when a user expands a tree branch by clicking the [+] button of an **Expandable** node. (Inherited from DiagramBase.)
	TreeItemTextEditing	A validation event that lets you approve or reject changes made to the text of a TreeViewNode's item. (Inherited from DiagramBase.)
	ValidateAnchorPoint	A validation event that lets you approve or reject connecting a link to an anchor point. (Inherited from DiagramBase.)
	ValidateLinkCrossing	A validation event that lets you approve or reject displaying a decoration at the intersection point of links. (Inherited from DiagramBase.)
	ViewRemoved	Raised when a DiagramView is detached from this diagram. (Inherited from DiagramBase.)

See Also

Diagram Class
MindFusion.Diagramming Namespace

Diagram Fields

Diagram.Dummy Field

A shared DummyNode instance.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DummyNode Dummy
```

Visual Basic Copy Code

```
Public Dummy As DummyNode
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram Constructor

The Diagram class represents a flow diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Diagram ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

Diagram Class
MindFusion.Diagramming Namespace

Diagram Methods

Diagram.AddItem Method

Invoked by the AddItemCmd and RemoveItemCmd classes to effectively add the specified item to the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void AddItem (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Protected Sub AddItem( _
    item As DiagramItem _
)
```

Parameters

item
The DiagramItem being added to the diagram.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
RemoveItem Method

Diagram.AlignPointToGrid Method

Returns a point of the alignment grid nearest to the one passed as an argument.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point AlignPointToGrid (
    Point point
)
```

Visual Basic Copy Code

```
Public Function AlignPointToGrid( _
    point As Point _
) As Point
```

Parameters

point
A point specified in diagram coordinates.

Return Value

A point from the alignment grid.

Remarks

Items created or moved by a user are automatically aligned to the grid. However, if items are placed in the diagram programmatically, their coordinates are set to the specified positions without aligning. In such cases use **AlignPointToGrid** if items must be placed exactly at grid points.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AlignToGrid Property
GridSizeX Property
GridSizeY Property

Diagram.AllowNodeBounds Method

Determines whether node can be placed at specified coordinates.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowNodeBounds (
    DiagramNode node,
    Rect rect
)
```

Visual Basic Copy Code

```
Public Function AllowNodeBounds( _
    node As DiagramNode, _
    rect As Rect _
) As Boolean
```

Parameters

node
A DiagramNode instance.

rect
A **Microsoft.Maui.Graphics.Rect** specifying node's coordinates.

Return Value

true if node and diagram constraints allow placing the node at specified coordinates, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.Arrange Method

Arranges the diagram using the specified graph layout algorithm.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Arrange (
    ILayout layout
)
```

Visual Basic Copy Code

```
Public Function Arrange( _
    layout As ILayout _
) As Boolean
```

Parameters

layout
An instance of Layout -derived class specifying the layout algorithm to apply.

Return Value

true if the diagram has been arranged successfully, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ArrangeAnimated Method

Arranges the diagram using the specified graph layout algorithm, and animates items moving to their new positions.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ArrangeAnimated (
    ILayout layout,
    int duration,
    AnimationType animationType,
    EasingType easingType
)
```

Visual Basic Copy Code

```
Public Function ArrangeAnimated( _
    layout As ILayout, _
    duration As Integer, _
    animationType As AnimationType, _
    easingType As EasingType _
) As Boolean
```

Parameters

layout
An instance of Layout -derived class specifying the layout algorithm to apply.

duration
An integer, specifying the duration of the animation in milliseconds.

animationType
A member of the AnimationType enumeration, specifying the animation type to use.

easingType
A member of the EasingType enumeration, specifying the type of easing function to apply to the animation.

Return Value

true if the diagram has been arranged successfully, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ArrangeAsync Method

Arranges the diagram asynchronously using the specified graph layout algorithm.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task<Boolean> ArrangeAsync (
    ILayout layout
)
```

Visual Basic Copy Code

```
Public Function ArrangeAsync( _
    layout As ILayout _
) As Task(Of Boolean)
```

Parameters

layout
An instance of Layout -derived class specifying the layout algorithm to apply.

Return Value

true if the diagram has been arranged successfully, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ArrangeLinkLabels Method

Rearranges link labels whose AutoArrange property is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ArrangeLinkLabels ()
```

Visual Basic Copy Code

```
Public Sub ArrangeLinkLabels ()
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LinkLabel Class

Diagram.ChangeUnit Method

Sets new MeasureUnit value and scales the coordinates of diagram items so that their absolute size remains unchanged.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ChangeUnit (
    MeasureUnit newUnit
)
```

Visual Basic Copy Code

```
Public Sub ChangeUnit( _
    newUnit As MeasureUnit _
)
```

Parameters

newUnit

A MeasureUnit instance specifying the new unit.

Remarks

This method scales the coordinates of nodes' **Bounds** values and links' ControlPoints to preserve their size when measured in the new unit. In addition, it scales values of some size and distance -related properties such as ShadowOffsetX and HeadShapeSize.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ClearAll Method

Removes all items from the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ClearAll ()
```

Visual Basic Copy Code

```
Public Sub ClearAll ()
```

Remarks

Use this method to clear all Groups, Links and Nodes in the diagram.

See Also

Diagram.Clone Method

Creates a copy of this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Diagram Clone ()
```

Visual Basic Copy Code

```
Public Overridable Function Clone () As Diagram
```

Return Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.CopySelection Method

Creates a copy of specified diagram elements.

Overload List

Name	Description
Diagram.CopySelection (Boolean, Boolean)	Creates a copy of the currently selected objects and groups.
Diagram.CopySelection (DiagramItemCollection, Boolean, Boolean)	Creates a copy of the specified objects and groups.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.CopySelection Method (Boolean, Boolean)

Creates a copy of the currently selected objects and groups.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SelectionCopy CopySelection (
    bool unconnectedLinks,
    bool copyGroups
)
```

Visual Basic Copy Code

```
Public Function CopySelection( _
    unconnectedLinks As Boolean, _
    copyGroups As Boolean _
) As SelectionCopy
```

Parameters

unconnectedLinks
true to allow copying unconnected links, or **false** otherwise.

copyGroups
true to copy group information, or **false** otherwise.

Return Value

An instance of the SelectionCopy class.

Remarks

Call PasteSelection to add the copied objects to a target diagram.

See Also

CopySelection Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.CopySelection Method (DiagramItemCollection, Boolean, Boolean)

Creates a copy of the specified objects and groups.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SelectionCopy CopySelection (
    DiagramItemCollection selectedItems,
    bool unconnectedLinks,
    bool copyGroups
)
```

Visual Basic Copy Code

```
Public Function CopySelection( _
    selectedItems As DiagramItemCollection, _
    unconnectedLinks As Boolean, _
    copyGroups As Boolean _
) As SelectionCopy
```

Parameters

selectedItems
A list of items to copy.

unconnectedLinks
true to allow copying unconnected links, or **false** otherwise.

copyGroups
true to copy group information, or **false** otherwise.

Return Value

An instance of the SelectionCopy class.

Remarks

Call PasteSelection to add the copied objects to a target diagram.

See Also

CopySelection Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.CopyTo Method

Copies this diagram's properties and items to specified target diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void CopyTo (
    Diagram target
)
```

Visual Basic Copy Code

```
Public Overridable Sub CopyTo( _
    target As Diagram _
)
```

Parameters

target
The target diagram.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.CreateAddItemCmd Method

Creates an AddItemCmd instance that implements undo/redo of adding an item to the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual AddItemCmd CreateAddItemCmd (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Protected Overridable Function CreateAddItemCmd( _
    item As DiagramItem _
) As AddItemCmd
```

Parameters

item
The item being added to the diagram.

Return Value

An instance of AddItemCmd or derived class.

Remarks

You can override this method to return a custom command that stores additional information about the added item, or handles the addition of more than one items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AddItem Method
RemoveItem Method
CreateRemoveItemCmd Method

Diagram.CreateImage Method

Creates a bitmap containing an image of the diagram.

Overload List

Name	Description
Diagram.CreateImage ()	Creates a bitmap image representing this diagram.
Diagram.CreateImage (Rect)	Creates a bitmap image containing specified diagram region.
Diagram.CreateImage (Rect, Double)	Creates a bitmap image representing this diagram using the specified scale factor.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.CreateImage Method ()

Creates a bitmap image representing this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage CreateImage ()
```

Visual Basic Copy Code

```
Public Function CreateImage () As IImage
```

Return Value

A **Microsoft.Maui.Graphics.IImage** instance.

Remarks

The method creates a new bitmap instance as big as the current document extents and paints the diagram in it. The **Save** method of the **IImage** object can be used to save the image in a jpeg, gif or bmp file.

See Also

CreateImage Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
HtmlBuilder Class

Diagram.CreateImage Method (Rect)

Creates a bitmap image containing specified diagram region.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage CreateImage (
    Rect bounds
)
```

Visual Basic Copy Code

```
Public Function CreateImage( _
    bounds As Rect _
) As IImage
```

Parameters

bounds
A Rect specifying what region of the diagram to export, in logical coordinates.

Return Value

An ImageSource instance.

See Also

CreateImage Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.CreateImage Method (Rect, Double)

Creates a bitmap image representing this diagram using the specified scale factor.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage CreateImage (
    Rect viewport,
    double scale
)
```

Visual Basic Copy Code

```
Public Function CreateImage( _
    viewport As Rect, _
    scale As Double _
) As IImage
```

Parameters

viewport
Specifies what region of the diagram to export.

scale
Specifies the scale factor as percentage of the original size.

Return Value

A **Microsoft.Maui.Graphics.IImage** instance containing the diagram image.

Remarks

The method creates a new bitmap instance as big as the current document extents and paints the diagram in it. The **Save** method of the **IImage** object can be used to save the image in a jpeg, gif or bmp file.

See Also

CreateImage Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
HtmlBuilder Class

Diagram.CreateMeasureGraphics Method

Creates an IGraphics object that can be used to obtain various string size measurements.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IGraphics CreateMeasureGraphics ()
```

Visual Basic Copy Code

```
Public Function CreateMeasureGraphics () As IGraphics
```

Return Value

An object implementing the IGraphics interface.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.CreateModifyItemCmd Method

Creates a ModifyItemCmd instance that implements undo/redo of item move and resize operations.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual ModifyItemCmd CreateModifyItemCmd (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Protected Overridable Function CreateModifyItemCmd( _
    item As DiagramItem _
) As ModifyItemCmd
```

Parameters

item
The item being modified.

Return Value

An instance of ModifyItemCmd or derived class.

Remarks

By overriding this method, you can use custom ModifyItemCmd objects to store additional information about the modified item, or handle the modification of additional items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.CreateRemoveItemCmd Method

Creates a RemoveItemCmd instance that implements undo/redo of removing an item from the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual RemoveItemCmd CreateRemoveItemCmd (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Protected Overridable Function CreateRemoveItemCmd( _
    item As DiagramItem _
) As RemoveItemCmd
```

Parameters

item
The item being removed from the diagram.

Return Value

An instance of RemoveItemCmd or derived class.

Remarks

You can override this method to return a custom command that stores additional information about the removed item, or handles the removal of more than one items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AddItem Method
RemoveItem Method
CreateAddItemCmd Method

Diagram.CreateThemeFromDefaults Method

Creates a Theme object representing the default properties of the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Theme CreateThemeFromDefaults ()
```

Visual Basic Copy Code

```
Public Function CreateThemeFromDefaults () As Theme
```

Return Value

An instance of the Theme class.

Remarks

This method is only for compatibility with files saved with versions earlier than 6.0.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.DeleteGroup Method

Removes the specified group from the group collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void DeleteGroup (
    Group group
)
```

Visual Basic Copy Code

```
Protected Sub DeleteGroup( _
    group As Group _
)
```

Parameters

group
The Group to be removed.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.Dispose Method

Clean up any resources being used.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void Dispose (
    bool disposing
)
```

Visual Basic Copy Code

```
Protected Overrides Sub Dispose( _
    disposing As Boolean _
)
```

Parameters

disposing
true to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.Draw Method

Renders the diagram on the specified IGraphics object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void Draw (
    IGraphics graphics,
    RenderOptions options,
    Rect clipRect,
    bool backBuffer
)
```

Visual Basic Copy Code

```
Public Overridable Sub Draw( _
    graphics As IGraphics, _
    options As RenderOptions, _
    clipRect As Rect, _
    backBuffer As Boolean _
)
```

Parameters

graphics
The IGraphics surface to draw the diagram on.

options
A RenderOptions object specifying options for the drawing operation.

clipRect
A **Microsoft.Maui.Graphics.Rect** specifying the current clipping rectangle.

backBuffer

Remarks

MvcDiagram calls this method to draw a representation of the diagram onto the specified graphics surface. You can override **Draw** if you need to render additional elements over what's already drawn by the Diagram object.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.DrawAlignmentGuides Method

Draws the current alignment guides.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void DrawAlignmentGuides (
    IGraphics graphics
)
```

Visual Basic Copy Code

```
Public Sub DrawAlignmentGuides( _
    graphics As IGraphics _
)
```

Parameters

graphics
An IGraphics drawing surface.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.DrawInteraction Method

Draws the items that the user is currently modifying.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void DrawInteraction (
    IGraphics graphics,
    RenderOptions options,
    Rect clipRect
)
```

Visual Basic Copy Code

```
Public Sub DrawInteraction( _
    graphics As IGraphics, _
    options As RenderOptions, _
    clipRect As Rect _
)
```

Parameters

graphics
The IGraphics surface to draw the items on.

options

A RenderOptions object specifying options for the drawing operation.

clipRect

A **Microsoft.Maui.Graphics.Rect** specifying the current clipping rectangle.

Remarks

This method draws the items that are currently being modified by the user.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.DrawInteractions Method

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void DrawInteractions (
    IGraphics graphics,
    RenderOptions options,
    Rect clipRect
)
```

Visual Basic Copy Code

```
Public Sub DrawInteractions( _
    graphics As IGraphics, _
    options As RenderOptions, _
    clipRect As Rect _
)
```

Parameters

graphics

options

clipRect

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.DrawLayer Method

Renders all diagram items to the specified graphics.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void DrawLayer (
    IGraphics graphics,
    Rect backgroundRect,
    int layerIndex,
```

```
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Sub DrawLayer( _
    graphics As IGraphics, _
    backgroundRect As Rect, _
    layerIndex As Integer, _
    options As RenderOptions _
)
```

Parameters

graphics

backgroundRect

layerIndex

options

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.DrawStyledText Method

Draws a formatted text string, the same way text is drawn in shape nodes whose EnableStyledText property is set to **true**. This method lets you display styled text in custom-drawn nodes or links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static void DrawStyledText (
    IGraphics graphics,
    string text,
    Font font,
    Brush textBrush,
    Rect textBounds,
    StringFormat format
)
```

Visual Basic Copy Code

```
Public Shared Sub DrawStyledText( _
    graphics As IGraphics, _
    text As String, _
    font As Font, _
    textBrush As Brush, _
    textBounds As Rect, _
    format As StringFormat _
)
```

Parameters

graphics
The IGraphics canvas on which to draw.

text
The text that should be drawn, including formatting tags.

font
The font used to draw the text.

textBrush
The brush used to draw the text.

textBounds
The bounds of the text layout rectangle.

format
Specifies the text alignment.

Remarks

You might use this method when you custom-draw shape nodes and still need to use the EnableStyledText rendering functionality of MvcDiagram.

Example

The following sample code illustrates how to manually render styled text.

C# Copy Code

```
private void diagram_DrawNode(object sender, DrawNodeEventArgs e)
{
    Diagram.DrawStyledText(e.Graphics, "<b>test</b> <i>test</i>",
        diagram.Font, Brushes.Black, e.Node.Bounds, StringFormat.GenericDefault);
}
```

Visual Basic Copy Code

```
Private Sub diagram_DrawNode(ByVal sender As Object, ByVal e As DrawNodeEventArgs) Handles
diagram.DrawNode

    Diagram.DrawStyledText(e.Graphics, "<b>test</b> <i>test</i>", _
        diagram.Font, Brushes.Black, e.Node.Bounds, StringFormat.GenericDefault)

End Sub
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DrawNode Event

Diagram.EffectiveTouchHitDistance Method

Returns TouchHitDistance or a default value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double EffectiveTouchHitDistance ()
```

Visual Basic Copy Code

```
Public Function EffectiveTouchHitDistance () As Double
```

Return Value

Distance from adjustment handles at which they are still considered as hit when processing touch input.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.EnumByReverseZ Method

Hit-tests diagram items in reverse Z order.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IEnumerable<DiagramItem> EnumByReverseZ (
    Point hitTest
)
```

Visual Basic Copy Code

```
Public Function EnumByReverseZ( _
    hitTest As Point _
) As IEnumerable(Of DiagramItem)
```

Parameters

hitTest
A point specifying at what location to search for items.

Return Value

The found items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.EnumTopLevelByReverseZ Method

Hit-tests diagram top-level items in reverse Z order.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IEnumerable<DiagramItem> EnumTopLevelByReverseZ (
    Point hitTest
)
```

Visual Basic Copy Code

```
Public Function EnumTopLevelByReverseZ( _
    hitTest As Point _
) As IEnumerable(Of DiagramItem)
```

Parameters

hitTest
A point specifying at what location to search for items.

Return Value

The found items.

See Also

Diagram.ExecuteCommand Method

Executes the specified command on this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ExecuteCommand (
    Command cmd
)
```

Visual Basic Copy Code

```
Public Sub ExecuteCommand( _
    cmd As Command _
)
```

Parameters

cmd
The Command to execute.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.FindGroup Method

Finds the group that has the specified tag value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Group FindGroup (
    Object tagValue
)
```

Visual Basic Copy Code

```
Public Function FindGroup( _
    tagValue As Object _
) As Group
```

Parameters

tagValue
A .NET object assigned as a tag and characterizing the group to be searched for.

Return Value

The Group instance if found, or **null** otherwise.

Remarks

Use this method to find a group that has been marked earlier with the specified tag.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
FindLink Method
FindNode Method

Diagram.FindLink Method

Finds the link that has the specified tag value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink FindLink (
    Object tagValue
)
```

Visual Basic Copy Code

```
Public Function FindLink( _
    tagValue As Object _
) As DiagramLink
```

Parameters

tagValue
A .NET object assigned as a tag and characterizing the link to be searched for.

Return Value

The DiagramLink instance if found, or **null** otherwise.

Remarks

Use this method to find a link that has been marked earlier with the specified tag.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
FindNode Method
FindGroup Method
FindLinkById Method

Diagram.FindLinkById Method

Finds the link that has the specified id value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink FindLinkById (
    Object id
)
```

Visual Basic Copy Code

```
Public Function FindLinkById( _
    id As Object _
) As DiagramLink
```

Parameters

id
A .NET object used as a link id.

Return Value

A DiagramLink whose Id equals to the specified value if such link exists, or **null** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
FindNodeById Method
FindLink Method

Diagram.FindNode Method

Finds the node that has the specified tag value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode FindNode (
    Object tagValue
)
```

Visual Basic Copy Code

```
Public Function FindNode( _
    tagValue As Object _
) As DiagramNode
```

Parameters

tagValue
A .NET object assigned as a tag and characterizing the node to be searched for.

Return Value

The DiagramNode instance if found, or **null** otherwise.

Remarks

Use this method to find a node that has been marked earlier with the specified tag.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
FindLink Method
FindGroup Method
FindNodeById Method

Diagram.FindNodeById Method

Finds the node that has the specified id value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode FindNodeById (
    Object id
)
```

Visual Basic Copy Code

```
Public Function FindNodeById( _
    id As Object _
) As DiagramNode
```

Parameters

id
A .NET object used as a node id.

Return Value

A DiagramNode whose Id equals to the specified value if such node exists, or **null** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
FindLinkById Method
FindNode Method

Diagram.GetContentBounds Method

Returns the smallest rectangle that bounds all diagram objects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect GetContentBounds (
    bool forPrint,
    bool onlyVisible
)
```

Visual Basic Copy Code

```
Public Function GetContentBounds( _
    forPrint As Boolean, _
    onlyVisible As Boolean _
) As Rect
```

Parameters

forPrint
Specifies whether only printable items should be considered.

onlyVisible
Specifies whether only visible items should be considered.

Return Value

The smallest rectangle that bounds all diagram items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.GetItemAt Method

Overload List

Name	Description
Diagram.GetItemAt (Point, Boolean)	Finds and returns the top-most diagram item that lies at the specified location.
Diagram.GetItemAt (Point, Predicate(DiagramItem))	Returns the top-most diagram item lying at the specified location.
Diagram.GetItemAt (Point, Predicate(DiagramItem), Boolean, Boolean, Boolean, Boolean)	Returns the top-most diagram item lying at the specified location.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.GetItemAt Method (Point, Boolean)

Finds and returns the top-most diagram item that lies at the specified location.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem GetItemAt (
    Point point,
    bool exclLocked
)
```

Visual Basic Copy Code

```
Public Function GetItemAt( _
    point As Point, _
    exclLocked As Boolean _
) As DiagramItem
```

Parameters

point
Specifies a location in the diagram where to look for items.

exclLocked
Specifies whether to ignore locked items.

Return Value

The DiagramItem found or **null** (**Nothing** in Visual Basic) if no item has been found.

Remarks

Call the method to get the item at a certain location in the diagram. The method searches starting from the top of z-order and returns the first item found.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetNodeAt Method

GetLinkAt Method
LinkHitDistance Property

Diagram.GetItemAt Method (Point, Predicate(DiagramItem))

Returns the top-most diagram item lying at the specified location.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem GetItemAt (
    Point point,
    Predicate<DiagramItem> predicate
)
```

Visual Basic Copy Code

```
Public Function GetItemAt( _
    point As Point, _
    predicate As Predicate(Of DiagramItem) _
) As DiagramItem
```

Parameters

point
Specifies a location in the diagram where to look for items.

predicate
A predicate specifying which items to hit-test.

Return Value

A DiagramItem instance found at the specified point.

See Also

GetItemAt Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.GetItemAt Method (Point, Predicate(DiagramItem), Boolean, Boolean, Boolean, Boolean)

Returns the top-most diagram item lying at the specified location.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem GetItemAt (
    Point point,
    Predicate<DiagramItem> predicate,
    bool exclLocked,
    bool includeHandles,
    bool includeLabels,
    bool activeFirst
)
```

Visual Basic Copy Code

```
Public Function GetItemAt( _
    point As Point, _
    predicate As Predicate(Of DiagramItem), _
    exclLocked As Boolean, _
    includeHandles As Boolean, _
    includeLabels As Boolean, _
    activeFirst As Boolean _
) As DiagramItem
```

Parameters

point
Specifies a location in the diagram where to look for items.

predicate
A predicate specifying which items to hit-test.

exclLocked
Specifies whether locked links should be excluded from the search.

includeHandles
true to also consider adjustment handle locations (possibly outside of item's bounds).

includeLabels
true to also consider item label locations (possibly outside of item's bounds).

activeFirst
Specifies whether to test ActiveItem first, disregarding Z order.

Return Value

A DiagramItem instance found at the specified point.

See Also

GetItemAt Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.GetLinkAt Method

Finds and returns the link that lies at the specified location.

Overload List

Name	Description
Diagram.GetLinkAt (Point, Double)	Finds and returns the top-most link that lies at the specified location.
Diagram.GetLinkAt (Point, Double, Boolean)	Finds and returns the top-most link that lies at the specified location, optionally excluding locked links from the search.
Diagram.GetLinkAt (Point, Double, Boolean, Int32@)	Finds and returns the top-most link that lies at the specified location, optionally excluding locked links from the search. The zero-based index of the exact link segment at the specified position is also returned.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetNodeAt Method
GetItemAt Method

Diagram.GetLinkAt Method (Point, Double)

Finds and returns the top-most link that lies at the specified location.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink GetLinkAt (
    Point point,
    double maxDist
)
```

Visual Basic Copy Code

```
Public Function GetLinkAt( _
    point As Point, _
    maxDist As Double _
) As DiagramLink
```

Parameters

point
A Point specifying a diagram point in logical coordinates.

maxDist
The maximal distance between the point and a link, within which to consider the point as lying on the link.

Return Value

The DiagramLink found or a **null** reference (**Nothing** in Visual Basic) if no item has been found.

Remarks

Call the method to get the link at a certain place in the document. The method searches starting from the top of z-order and returns the first link found. The specified point might not be laying exactly on the link, but within a certain threshold distance from it. This threshold is specified through the second parameter.

See Also

GetLinkAt Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetNodeAt Method
GetItemAt Method
GetLinksAt Method

Diagram.GetLinkAt Method (Point, Double, Boolean)

Finds and returns the top-most link that lies at the specified location, optionally excluding locked links from the search.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink GetLinkAt (
    Point point,
    double maxDist,
    bool exclLocked
)
```

Visual Basic Copy Code

```
Public Function GetLinkAt( _
    point As Point, _
    maxDist As Double, _
    exclLocked As Boolean _
) As DiagramLink
```

Parameters

point
A Point specifying a diagram point in logical coordinates.

maxDist
The maximal distance between the point and a link, within which to consider the point as lying on the link.

exclLocked
Specifies that locked links should be ignored by the search.

Return Value

The DiagramLink found or a **null** reference (**Nothing** in Visual Basic) if no item has been found.

Remarks

Call the method to get the link at a certain place in the document. The method searches starting from the top of z-order and returns the first link found. The specified point might not be laying exactly on the link, but within a certain threshold distance from it. This threshold is specified through the second parameter. Locked and invisible links can be skipped from the search by setting the third parameter to **true**.

See Also

GetLinkAt Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetItemAt Method
GetNodeAt Method

Diagram.GetLinkAt Method (Point, Double, Boolean, Int32@)

Finds and returns the top-most link that lies at the specified location, optionally excluding locked links from the search. The zero-based index of the exact link segment at the specified position is also returned.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink GetLinkAt (
    Point point,
    double maxDist,
    bool exclLocked,
    ref int segmNum
)
```

Visual Basic Copy Code

```
Public Function GetLinkAt( _
    point As Point, _
    maxDist As Double, _
    exclLocked As Boolean, _
    ByRef segmNum As Integer _
) As DiagramLink
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying a diagram point in logical coordinates.

maxDist
The maximal distance between the point and a link, within which to consider the point as lying on the link.

exclLocked
Specifies that locked links should be ignored by the search.

segmNum
Returns the index of the link segment that lies at the specified point.

Return Value

The DiagramLink found or a **null** reference (**Nothing** in Visual Basic) if no item has been found.

Remarks

Call the method to get the link at a certain place in the document. The method searches starting from the top of z-order and returns the first link found. The specified point might not be laying exactly on the link, but within a certain threshold distance from it. This threshold is specified through the second parameter. Locked and invisible links can be skipped from the search by setting the third parameter to **true**. The link's segment closest to the point is returned through the fourth parameter.

See Also

GetLinkAt Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetNodeAt Method
GetItemAt Method

Diagram.GetLinksAt Method

Returns the links that contain the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection GetLinksAt (
    Point point
)
```

Visual Basic Copy Code

```
Public Function GetLinksAt( _
    point As Point _
) As DiagramLinkCollection
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** structure specifying the location at which to look for links.

Return Value

A DiagramLinkCollection instance containing the links lying at the specified location.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetLinkAt Method
GetNodesAt Method

Diagram.GetNearestNode Method

Returns the node nearest to the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode GetNearestNode (
    Point point,
    double maxDistance,
```

```
    DiagramNode ignored
)
```

Visual Basic Copy Code

```
Public Function GetNearestNode( _
    point As Point, _
    maxDistance As Double, _
    ignored As DiagramNode _
) As DiagramNode
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** instance identifying a diagram location.

maxDistance
The maximum distance from the point at which to look for nodes.

ignored
A node that should be ignored.

Return Value

The DiagramNode nearest to the specified point.

Remarks

The last argument is useful when calling this method from NodeCreated and NodeModified handlers and the affected node should not be included in the search.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetNodeAt Method

Diagram.GetNodeAt Method

Gets the node that lies at the specified location.

Overload List

Name	Description
Diagram.GetNodeAt (Point)	Returns the topmost node that lies at the specified position.
Diagram.GetNodeAt (Point, Boolean)	Returns the topmost node that lies at the specified position.
Diagram.GetNodeAt (Point, Boolean, Boolean)	Finds and returns the top-most node that lies at the specified location, optionally excluding locked and/or selected nodes from the search.
Diagram.GetNodeAt (Point, Boolean, Boolean, Double)	Finds and returns the top-most node that lies within the specified threshold from the specified location, optionally excluding locked and/or selected nodes from the search.
Diagram.GetNodeAt (Point, Double)	Finds and returns the top-most node that lies within the specified threshold from the specified location.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetItemAt Method
GetLinkAt Method

Diagram.GetNodeAt Method (Point)

Returns the topmost node that lies at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode GetNodeAt (
    Point point
)
```

Visual Basic Copy Code

```
Public Function GetNodeAt( _
    point As Point _
) As DiagramNode
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** instance identifying a diagram location.

Return Value

The DiagramNode found or a **null** reference (**Nothing** in Visual Basic) if no item has been found.

Remarks

The method starts searching from the top of the z-order and returns the first node that contains the specified point.

See Also

GetNodeAt Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetLinkAt Method
GetItemAt Method
GetNearestNode Method

Diagram.GetNodeAt Method (Point, Boolean)

Returns the topmost node that lies at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode GetNodeAt (
    Point point,
    bool excludeLocked
)
```

Visual Basic Copy Code

```
Public Function GetNodeAt( _
    point As Point, _
    excludeLocked As Boolean _
) As DiagramNode
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying a diagram point in logical coordinates.

excludeLocked
Specifies whether locked nodes should be excluded from the search.

Return Value

A DiagramNode instance if a node lies at the specified location, otherwise **null**.

See Also

GetNodeAt Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.GetNodeAt Method (Point, Boolean, Boolean)

Finds and returns the top-most node that lies at the specified location, optionally excluding locked and/or selected nodes from the search.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode GetNodeAt (
    Point point,
    bool excludeLocked,
    bool excludeSelected
)
```

Visual Basic Copy Code

```
Public Function GetNodeAt( _
    point As Point, _
    excludeLocked As Boolean, _
    excludeSelected As Boolean _
) As DiagramNode
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** instance identifying a diagram location.

excludeLocked
true if nodes that are locked should be omitted by searching; otherwise, **false**.

excludeSelected
true if nodes that are selected should be omitted by searching; otherwise, **false**.

Return Value

The DiagramNode found or a **null** reference (**Nothing** in Visual Basic) if no node has been found.

Remarks

This method gives you the possibility to specify search options when looking for a node occupying some location in the diagram document. The method searches starting from the top of z-order and returns the first object found.

See Also

GetNodeAt Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetItemAt Method
GetLinkAt Method
GetNearestNode Method

Diagram.GetNodeAt Method (Point, Boolean, Boolean, Double)

Finds and returns the top-most node that lies within the specified threshold from the specified location, optionally excluding locked and/or selected nodes from the search.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode GetNodeAt (
    Point point,
    bool excludeLocked,
    bool excludeSelected,
    double threshold
)
```

Visual Basic Copy Code

```
Public Function GetNodeAt( _
    point As Point, _
    excludeLocked As Boolean, _
    excludeSelected As Boolean, _
    threshold As Double _
) As DiagramNode
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** instance identifying a diagram location.

excludeLocked
true if nodes that are locked should be omitted by searching; otherwise, **false**.

excludeSelected
true if nodes that are selected should be omitted by searching; otherwise, **false**.

threshold
Allowed distance from the node.

Return Value

The DiagramNode found or a **null** reference (**Nothing** in Visual Basic) if no node has been found.

Remarks

This method gives you the possibility to specify search options when looking for a node occupying some location in the diagram document. The method searches starting from the top of z-order and returns the first object found.

See Also

GetNodeAt Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetLinkAt Method
GetItemAt Method
GetNearestNode Method

Diagram.GetNodeAt Method (Point, Double)

Finds and returns the top-most node that lies within the specified threshold from the specified location.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode GetNodeAt (
    Point point,
    double threshold
)
```

Visual Basic Copy Code

```
Public Function GetNodeAt( _
    point As Point, _
    threshold As Double _
) As DiagramNode
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** instance identifying a diagram location.

threshold
Allowed distance from the node.

Return Value

The DiagramNode found or a **null** reference (**Nothing** in Visual Basic) if no node has been found.

Remarks

This method gives you the possibility to specify search options when looking for a node occupying some location in the diagram document. The method searches starting from the top of z-order and returns the first object found.

See Also

GetNodeAt Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetItemAt Method
GetLinkAt Method
GetNearestNode Method

Diagram.GetNodesAt Method

Returns the nodes that contain the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeCollection GetNodesAt (
    Point point
)
```

Visual Basic Copy Code

```
Public Function GetNodesAt( _
    point As Point _
) As DiagramNodeCollection
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** instance identifying a diagram location.

Return Value

An instance of the DiagramNodeCollection class containing all found nodes; **null** (**Nothing** in Visual Basic) if no node has been found.

Remarks

This method gives you the possibility to specify search options when looking for the nodes occupying some location in the diagram document.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
GetItemAt Method
GetLinksAt Method

Diagram.GetNodesInViewport Method

Returns all nodes visible in the current viewport.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeCollection GetNodesInViewport ()
```

Visual Basic Copy Code

```
Public Function GetNodesInViewport () As DiagramNodeCollection
```

Return Value

A DiagramNodeCollection that contains the viewport nodes.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.GetValue Method

Gets the effective value of the style property with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected Object GetValue (
    string name
)
```

Visual Basic Copy Code

```
Protected Function GetValue( _
    name As String _
) As Object
```

Parameters

name
The name of the property to get.

Return Value

An object representing the effective property value.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.HandleKeyDown Method

Propagates keyboard input to interested manipulators and components.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool HandleKeyDown (
    int keyCode,
    int modifiers
)
```

Visual Basic Copy Code

```
Public Function HandleKeyDown( _
    keyCode As Integer, _
    modifiers As Integer _
) As Boolean
```

Parameters

keyCode
An integer value representing the key that was pressed.

modifiers
An integer value representing the modifier keys that were pressed.

Return Value

true if a component or manipulator handled the event; otherwise, **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.HandleKeyPress Method

Propagates keyboard input to interested manipulators.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void HandleKeyPress (
    char keyChar
)
```

Visual Basic Copy Code

```
Public Sub HandleKeyPress( _
    keyChar As Char _
)
```

Parameters

keyChar
The character representation of the pressed key.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.HandleKeyUp Method

Propagates keyboard input to interested components.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void HandleKeyUp (
    int keyCode,
    int modifiers
)
```

Visual Basic Copy Code

```
Public Sub HandleKeyUp( _
    keyCode As Integer, _
    modifiers As Integer _
)
```

Parameters

keyCode
An integer value representing the key that was pressed.

modifiers
An integer value representing the modifier keys that were pressed.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.HandleMouseMove Method

Propagates mouse input to interested manipulators.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CursorHint HandleMouseMove (
    Point point
)
```

Visual Basic Copy Code

```
Public Function HandleMouseMove( _
    point As Point _
) As CursorHint
```

Parameters

point

Return Value

A member of the CursorHint enumeration specifying the cursor to display at the specified point.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.HitTestComponents Method

Tests whether the specified point is within a component of a diagram item and returns this component.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ComponentBase HitTestComponents (
    Point point
)
```

Visual Basic Copy Code

```
Public Function HitTestComponents( _
    point As Point _
) As ComponentBase
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** representing the point to test, expressed in document coordinates.

Return Value

A ComponentBase instance representing the topmost component at the specified point.

Remarks

This method ignores components with any of the following properties set to **false**: IsHitTestVisible, IsEnabled, IsInteractive.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.HitTestLabels Method

Returns the item label located at specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ItemLabel HitTestLabels (
    Point point
)
```

Visual Basic Copy Code

```
Public Function HitTestLabels( _
    point As Point _
) As ItemLabel
```

Parameters

point
A Microsoft**.Maui.Graphics.Point** specifying where to look for labels.

Return Value

An ItemLabel instance, or **null** if there was no label found.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LinkLabel Class
NodeLabel Class

Diagram.HitTestManipulators Method

Tests whether the specified point is within the manipulator of a diagram object and triggers the manipulator.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Manipulator HitTestManipulators (
    Point point
)
```

Visual Basic Copy Code

```
Public Function HitTestManipulators( _
    point As Point _
) As Manipulator
```

Parameters

point

Return Value

true to indicate whether there is a manipulator at the specified location; otherwise, **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.Invalidate Method

Invalidates the diagram or a region of the diagram, causing it to be repainted.

Overload List

Name	Description
Diagram.Invalidate ()	Invalidates the entire diagram.

Diagram.Invalidate (Boolean)	Invalidates the entire diagram and optionally repaints the back-buffer image.
Diagram.Invalidate (Object, Rect)	Invalidates the specified rectangle of the diagram, causing it to be repainted.
Diagram.Invalidate (Rect)	Invalidates the specified rectangle of the diagram, causing it to be repainted.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramView.RecreateCacheImage Method

Diagram.Invalidate Method ()

Invalidates the entire diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Invalidate ()
```

Visual Basic Copy Code

```
Public Sub Invalidate ()
```

Remarks

This method raises the Repaint event. **DiagramView** objects handle **Repaint** by redrawing the specified diagram region.

See Also

Invalidate Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramView.RecreateCacheImage Method

Diagram.Invalidate Method (Boolean)

Invalidates the entire diagram and optionally repaints the back-buffer image.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Invalidate (
    bool redrawCacheImage
)
```

Visual Basic Copy Code

```
Public Sub Invalidate( _
    redrawCacheImage As Boolean _
)
```

Parameters

redrawCacheImage
A boolean value specifying whether the diagram cache image used during modification should be re-created.

Remarks

This method raises the Repaint event. **DiagramView** objects handle **Repaint** by redrawing the specified diagram region.

See Also

Invalidate Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramView.RecreateCacheImage Method

Diagram.Invalidate Method (Object, Rect)

Invalidates the specified rectangle of the diagram, causing it to be repainted.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Invalidate (
    Object sender,
    Rect invalidRect
)
```

Visual Basic Copy Code

```
Public Sub Invalidate( _
    sender As Object, _
    invalidRect As Rect _
)
```

Parameters

sender
The object that causes the diagram invalidation.

invalidRect
A **Microsoft.Maui.Graphics.Rect** instance specifying the region that should be repainted. If not specified, the whole diagram will be repainted.

Remarks

This method raises the Repaint event. **DiagramView** objects handle **Repaint** by redrawing the specified diagram region.

See Also

Invalidate Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramView.RecreateCacheImage Method

Diagram.Invalidate Method (Rect)

Invalidates the specified rectangle of the diagram, causing it to be repainted.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Invalidate (
    Rect invalidRect
)
```

Visual Basic Copy Code

```
Public Sub Invalidate( _
    invalidRect As Rect _
)
```

Parameters

invalidRect

A **Microsoft.Maui.Graphics.Rect** instance specifying the region that should be repainted. If not specified, the whole diagram will be repainted.

Remarks

This method raises the Repaint event. **DiagramView** objects handle **Repaint** by redrawing the specified diagram region.

See Also

Invalidate Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramView.RecreateCacheImage Method

Diagram.InvalidateZOrder Method

Forces a new sort of the Z order list before next redraw.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void InvalidateZOrder ()
```

Visual Basic Copy Code

```
Public Sub InvalidateZOrder ()
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.IsClassRegistered Method

Checks if a class is registered for serialization support.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static bool IsClassRegistered (
    Type classType
)
```

Visual Basic

```
Public Shared Function IsClassRegistered( _
    classType As Type _
) As Boolean
```

Parameters

classType
A Type instance identifying the class.

Return Value

true if the class is registered, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.IsClientClassRegistered Method

Checks if a class is registered for JSON serialization support.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static bool IsClientClassRegistered (
    Type classType
)
```

Visual Basic

```
Public Shared Function IsClientClassRegistered( _
    classType As Type _
) As Boolean
```

Parameters

classType
A Type instance identifying the class.

Return Value

true if the class is registered, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.IsItemInteractive Method

Checks if the specified item can be modified interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool IsItemInteractive (
    DiagramItem item
```

```
)
```

Visual Basic Copy Code

```
Public Function IsItemInteractive( _
    item As DiagramItem _
) As Boolean
```

Parameters

item
The DiagramItem to check.

Return Value

true, if the item is currently visible and unlocked and it is not associated with a hidden or locked layer; otherwise, **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.IsItemLocked Method

Checks if the specified item is locked.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsItemLocked (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Function IsItemLocked( _
    item As DiagramItem _
) As Boolean
```

Parameters

item
The DiagramItem to check.

Return Value

true, if the item's Locked property is set to true or if the item is associated with a locked layer; otherwise, **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.IsItemVisible Method

Checks if the specified item is currently visible.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsItemVisible (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Function IsItemVisible( _
    item As DiagramItem _
) As Boolean
```

Parameters

item
The DiagramItem to check.

Return Value

true if the item's Visible property is set to true and the item is not associated with a hidden layer; otherwise, **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.LoadFromJson Method

Loads the diagram from a JSON-formatted string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void LoadFromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Overridable Sub LoadFromJson( _
    json As String _
)
```

Parameters

json
A string created by the SaveToJson method.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
SaveToJson Method

Diagram.LoadFromJsonAsync Method

Loads the diagram from a JSON-formatted string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task LoadFromJsonAsync (
    string json
)
```

Visual Basic Copy Code

```
Public Function LoadFromJsonAsync( _
    json As String _
) As Task
```

Parameters

json

Return Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.LoadFromJsonFile Method

Loads the diagram from a JSON file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromJsonFile (
    string fileName
)
```

Visual Basic Copy Code

```
Public Sub LoadFromJsonFile( _
    fileName As String _
)
```

Parameters

fileName
A string specifying the file name or full path.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
SaveToJsonFile Method

Diagram.LoadFromJsonFileAsync Method

Loads the diagram from a JSON file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task LoadFromJsonFileAsync (
    string fileName
)
```

Visual Basic Copy Code

```
Public Function LoadFromJsonFileAsync( _
    fileName As String _
) As Task
```

Parameters

fileName

Return Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.LoadFromJsonObject Method

Loads the diagram from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void LoadFromJsonObject (
    JsonObject diagramObj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Public Overridable Sub LoadFromJsonObject( _
    diagramObj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

diagramObj
A JsonObject containing diagram data.

context
A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.LoadFromString Method

Decodes diagram contents from a string that was created with SaveToString.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromString (
    string str,
    SaveToStringFormat format
)
```

Visual Basic Copy Code

```
Public Sub LoadFromString( _
    str As String, _
    format As SaveToStringFormat _
)
```

Parameters

str
A string containing the serialized diagram contents.

format

Remarks

Loads a diagram document from a string. The string must have been created by the SaveToString method.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
SaveToStream
LoadFromFile
SerializationBinder Property

Diagram.LoadFromXml Method

Loads the contents of an XML diagram document into the control.

Overload List

Name	Description
Diagram.LoadFromXml (Stream)	Loads the diagram from the specified XML stream.
Diagram.LoadFromXml (String)	Loads the contents of the XML diagram document from the specified file into the control.
Diagram.LoadFromXml (String, Boolean)	Loads the diagram from an XML file.
Diagram.LoadFromXml (XmlDocument)	Loads the contents of the XML diagram document represented by the specified **XmlDocument** into the control.
Diagram.LoadFromXml (XmlDocument, Boolean)	Loads the diagram from an XML document.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LoadFromFile Method
SaveToXml Method

Diagram.LoadFromXml Method (Stream)

Loads the diagram from the specified XML stream.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    Stream stream
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    stream As Stream _
)
```

Parameters

stream

See Also

LoadFromXml Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.LoadFromXml Method (String)

Loads the contents of the XML diagram document from the specified file into the control.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    string fileName
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    fileName As String _
)
```

Parameters

fileName
The name of the XML file the data should be read from.

Remarks

The method reads a diagram previously saved via the SaveToXml method. This method is implemented by loading an **XmlDocument** from the specified file and calling the LoadFromXml overload.

See Also

LoadFromXml Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
SaveToXml Method

Diagram.LoadFromXml Method (String, Boolean)

Loads the diagram from an XML file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    string fileName,
    bool loadDiagramProperties
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    fileName As String, _
    loadDiagramProperties As Boolean _
)
```

Parameters

fileName
The name of the XML file the data should be read from.

loadDiagramProperties
true to load values of the diagram properties; **false** to load only the diagram items.

Remarks

The method reads a diagram previously saved via the SaveToXml method. This method is implemented by loading an **XmlDocument** from the specified file and calling the LoadFromXml overload.

See Also

LoadFromXml Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.LoadFromXml Method (XmlDocument)

Loads the contents of the XML diagram document represented by the specified **XmlDocument** into the control.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void LoadFromXml (
    XmlDocument document
)
```

Visual Basic Copy Code

```
Public Overridable Sub LoadFromXml( _
    document As XmlDocument _
)
```

Parameters

document
A .NET **XmlDocument** object from which to load the diagram data.

Remarks

The method reads a diagram previously saved via the SaveToXml method.

See Also

LoadFromXml Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
SaveToXml Method

Diagram.LoadFromXml Method (XmlDocument, Boolean)

Loads the diagram from an XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void LoadFromXml (
    XmlDocument document,
    bool loadDiagramProperties
)
```

Visual Basic Copy Code

```
Public Overridable Sub LoadFromXml( _
    document As XmlDocument, _
    loadDiagramProperties As Boolean _
)
```

Parameters

document
A .NET **XmlDocument** object from which to load the diagram data.

loadDiagramProperties
true to load values of the diagram properties; **false** to load only the diagram items.

Remarks

The method reads a diagram previously saved via the SaveToXml method.

See Also

LoadFromXml Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.LoadFromXmlAsync Method

Overload List

Name	Description
Diagram.LoadFromXmlAsync (String)	Loads the diagram from an XML file.
Diagram.LoadFromXmlAsync (XmlDocument)	Loads the diagram from an XML document.

See Also

Diagram.LoadFromXmlAsync Method (String)

Loads the diagram from an XML file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task LoadFromXmlAsync (
    string fileName
)
```

Visual Basic Copy Code

```
Public Function LoadFromXmlAsync( _
    fileName As String _
) As Task
```

Parameters

fileName

Return Value

See Also

LoadFromXmlAsync Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.LoadFromXmlAsync Method (XmlDocument)

Loads the diagram from an XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task LoadFromXmlAsync (
    XmlDocument document
)
```

Visual Basic Copy Code

```
Public Function LoadFromXmlAsync( _
    document As XmlDocument _
) As Task
```

Parameters

document

Return Value

See Also

Diagram.LoadSelectionFromJson Method

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadSelectionFromJson (
    string json,
    float offsetX,
    float offsetY,
    bool unconnectedLinks
)
```

Visual Basic Copy Code

```
Public Sub LoadSelectionFromJson( _
    json As String, _
    offsetX As Single, _
    offsetY As Single, _
    unconnectedLinks As Boolean _
)
```

Parameters

json

offsetX

offsetY

unconnectedLinks

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.MeasureString Method

Measures a text taking into consideration the current affine transform of the diagram.

Overload List

Name	Description
Diagram.MeasureString (String, Font, Int32, StringFormat)	Measures the specified text when drawn with the specified font in an area with the specified width taking into consideration the current affine transform of the diagram.
Diagram.MeasureString (String, Font, Rect, StringFormat, Int32@, Int32@)	Measures the specified text when drawn with the specified font in the specified rectangle area taking into consideration the current affine transform of the diagram.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.MeasureString Method (String, Font, Int32, StringFormat)

Measures the specified text when drawn with the specified font in an area with the specified width taking into consideration the current affine transform of the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size MeasureString (
    string text,
    Font font,
    int maxWidth,
    StringFormat textFormat
)
```

Visual Basic Copy Code

```
Public Function MeasureString( _
    text As String, _
    font As Font, _
    maxWidth As Integer, _
    textFormat As StringFormat _
) As Size
```

Parameters

text
The string to measure.

font
The font of the string.

maxWidth
The maximum width of the string.

textFormat
A StringFormat object that represents formatting information, such as line spacing, for the string.

Return Value

This method returns a **Microsoft.Maui.Graphics.Size** structure that represents the size of the string *text* when drawn with the *font* parameter and the *stringFormat* parameter.

See Also

MeasureString Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.MeasureString Method (String, Font, Rect, StringFormat, Int32@, Int32@)

Measures the specified text when drawn with the specified font in the specified rectangle area taking into consideration the current affine transform of the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size MeasureString (
    string text,
    Font font,
    Rect layoutArea,
    StringFormat stringFormat,
```

```
    out int charactersFitted,
    out int linesFilled
)
```

Visual Basic Copy Code

```
Public Function MeasureString( _
    text As String, _
    font As Font, _
    layoutArea As Rect, _
    stringFormat As StringFormat, _
    <System.Runtime.InteropServices.Out()> ByRef charactersFitted As Integer, _
    <System.Runtime.InteropServices.Out()> ByRef linesFilled As Integer _
) As Size
```

Parameters

text

The string to measure.

font

The font of the string.

layoutArea

A **Microsoft.Maui.Graphics.Rect** that specifies the maximum layout area for the text.

stringFormat

A **StringFormat** object that represents formatting information, such as line spacing, for the string.

charactersFitted

linesFilled

Return Value

This method returns a **Microsoft.Maui.Graphics.Size** that represents the size of the string *text* when drawn with the *font* parameter and the *stringFormat* parameter.

See Also

MeasureString Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.Navigate Method

Selects the nearest item in the specified direction.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Navigate (
    Point directionVector
)
```

Visual Basic Copy Code

```
Public Sub Navigate( _
    directionVector As Point _
)
```

Parameters

directionVector

A **Microsoft.Maui.Graphics.Point** object specifying the direction vector.

Remarks

This method can be used to let users navigate through the diagram using the keyboard keys.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.NewInstance Method

Custom Diagram classes must override this method to return an instance of their type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Diagram NewInstance ()
```

Visual Basic Copy Code

```
Public Overridable Function NewInstance () As Diagram
```

Return Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.PasteSelection Method

Adds the specified items and groups to the specified diagram document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool PasteSelection (
    SelectionCopy data,
    CompositeCmd cmd,
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Public Function PasteSelection( _
    data As SelectionCopy, _
    cmd As CompositeCmd, _
    dx As Double, _
    dy As Double _
) As Boolean
```

Parameters

data
A SelectionCopy object returned by a previous call to CopySelection.

cmd
A CompositeCmd object where the items state should be recorded for later undo or redo.

dx
Horizontal offset of pasted items from their original positions.

dy
Vertical offset of pasted items from their original positions.

Return Value

true if data contains any items; otherwise, **false**.

Remarks

This method can be used in conjunction with CopySelection to copy selected items from one Diagram instance to another.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
CopySelection

Diagram.RaiseCellTextEdited Method

Raises the CellTextEdited event .

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaiseCellTextEdited (
    TableNode.Cell cell,
    string oldText,
    string newText,
    int column,
    int row
)
```

Visual Basic Copy Code

```
Public Sub RaiseCellTextEdited( _
    cell As TableNode.Cell, _
    oldText As String, _
    newText As String, _
    column As Integer, _
    row As Integer _
)
```

Parameters

cell
The TableNode.Cell related to the event.

oldText
The old text of the cell.

newText
The new text of the cell.

column
The zero-based index of the table column related to the event.

row
The zero-based index of the table row related to the event.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseCellTextEditing Method

Raises the CellTextEditing event to validate entering in-place editing mode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RaiseCellTextEditing (
    TableNode table,
    int column,
    int row
)
```

Visual Basic Copy Code

```
Public Function RaiseCellTextEditing( _
    table As TableNode, _
    column As Integer, _
    row As Integer _
) As Boolean
```

Parameters

table
The TableNode related to the event.

column
The zero-based index of the table column related to the event.

row
The zero-based index of the table row related to the event.

Return Value

true if the in-place editing operation has been confirmed; otherwise, **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseClicked Method

Raises the Clicked event of the diagram or one of the *Clicked events related to diagram items.

Overload List

Name	Description
Diagram.RaiseClicked (DiagramItem, Point, MouseButton)	Raises the appropriate *Clicked event according to the type of the specified item.
Diagram.RaiseClicked (MouseButton, Point)	Raises the **Clicked** event of the Diagram class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseClicked Method (DiagramItem, Point, MouseButton)

Raises the appropriate *Clicked event according to the type of the specified item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaiseClicked (
    DiagramItem item,
    Point mousePosition,
    MouseButton mouseButton
)
```

Visual Basic Copy Code

```
Public Sub RaiseClicked( _
    item As DiagramItem, _
    mousePosition As Point, _
    mouseButton As MouseButton _
)
```

Parameters

item
The DiagramItem related to the event.

mousePosition
The current mouse position, in document coordinates.

mouseButton
The mouse button related to the event.

See Also

RaiseClicked Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseClicked Method (MouseButton, Point)

Raises the Clicked event of the Diagram class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaiseClicked (
    MouseButton mouseButton,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public Sub RaiseClicked( _
    mouseButton As MouseButton, _
    mousePosition As Point _
)
```

Parameters

mouseButton
The mouse button related to the event.

mousePosition
The current cursor position, in document coordinates.

See Also

RaiseClicked Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseCreated Method

Raises the appropriate *Created event according to the type of the specified item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaiseCreated (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Sub RaiseCreated( _
    item As DiagramItem _
)
```

Parameters

item
The DiagramItem related to the event.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseDeleting Method

Raises the appropriate *Deleting event according to the type of the specified item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RaiseDeleting (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Function RaiseDeleting( _
    item As DiagramItem _
) As Boolean
```

Parameters

item
The DiagramItem related to the event.

Return Value

true if the operation has been confirmed; otherwise, **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseDoubleClicked Method

Raises the DoubleClicked event of the diagram or one of the *DoubleClicked events related to diagram items.

Overload List

Name	Description
Diagram.RaiseDoubleClicked (DiagramItem, Point, MouseButton)	Raises the appropriate *DoubleClicked event according to the type of the specified item.
Diagram.RaiseDoubleClicked (MouseButton, Point)	Raises the **DoubleClicked** event of the Diagram class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseDoubleClicked Method (DiagramItem, Point, MouseButton)

Raises the appropriate *DoubleClicked event according to the type of the specified item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaiseDoubleClicked (
    DiagramItem item,
    Point mousePosition,
    MouseButton mouseButton
)
```

Visual Basic Copy Code

```
Public Sub RaiseDoubleClicked( _
    item As DiagramItem, _
    mousePosition As Point, _
    mouseButton As MouseButton _
)
```

Parameters

item
The DiagramItem related to the event.

mousePosition
The current cursor position, in document coordinates.

mouseButton
The mouse button related to the event.

See Also

RaiseDoubleClicked Method Overload List
Diagram Members

Diagram.RaiseDoubleClicked Method (MouseButton, Point)

Raises the DoubleClicked event of the Diagram class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaiseDoubleClicked (
    MouseButton mouseButton,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public Sub RaiseDoubleClicked( _
    mouseButton As MouseButton, _
    mousePosition As Point _
)
```

Parameters

mouseButton
The mouse button related to the event.

mousePosition
The current cursor position, in document coordinates.

See Also

RaiseDoubleClicked Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseLinkTextEdited Method

Raises the LinkTextEdited event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaiseLinkTextEdited (
    DiagramLink link,
    string oldText,
    string newText,
    LinkLabel label
)
```

Visual Basic Copy Code

```
Public Sub RaiseLinkTextEdited( _
    link As DiagramLink, _
    oldText As String, _
    newText As String, _
    label As LinkLabel _
)
```

Parameters

link
The DiagramLink related to the event.

oldText
The old text of the link.

newText
The new text of the link.

label
The LinkLabel related to the event.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseLinkTextEditing Method

Raises the LinkTextEditing event to validate entering inplace edit mode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RaiseLinkTextEditing (
    DiagramLink link,
    LinkLabel label
)
```

Visual Basic Copy Code

```
Public Function RaiseLinkTextEditing( _
    link As DiagramLink, _
    label As LinkLabel _
) As Boolean
```

Parameters

link
The DiagramLink related to the event.

label
The LinkLabel related to the event.

Return Value

true to let users edit the link's text, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseModified Method

Raises the NodeModified event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaiseModified (
    DiagramNode node,
    Point mousePosition,
    AdjustmentHandles adjustmentHandle
)
```

Visual Basic Copy Code

```
Public Sub RaiseModified( _
    node As DiagramNode, _
    mousePosition As Point, _
    adjustmentHandle As AdjustmentHandles _
)
```

Parameters

node

mousePosition

adjustmentHandle

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseMouseHover Method

Raises the appropriate Pointed event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void RaiseMouseHover (
    DiagramItem item,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public Overridable Sub RaiseMouseHover( _
    item As DiagramItem, _
    mousePosition As Point _
)
```

Parameters

item

mousePosition

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseNodeTextEdited Method

Raises the NodeTextEdited event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaiseNodeTextEdited (
    DiagramNode node,
    string oldText,
    string newText,
    NodeLabel label
)
```

Visual Basic Copy Code

```
Public Sub RaiseNodeTextEdited( _
    node As DiagramNode, _
    oldText As String, _
    newText As String, _
    label As NodeLabel _
)
```

Parameters

node
The DiagramNode related to the event.

oldText
The old text.

newText
The new text.

label
The NodeLabel related to the event.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseNodeTextEditing Method

Raises the NodeTextEditing event to validate entering inplace edit mode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RaiseNodeTextEditing (
    DiagramNode node,
    NodeLabel label = null
)
```

Visual Basic Copy Code

```
Public Function RaiseNodeTextEditing( _
    node As DiagramNode, _
    Optional label As NodeLabel = Nothing _
) As Boolean
```

Parameters

node
The DiagramNode related to the event.

label
Optional.
The NodeLabel related to the event.

Return Value

true if the in-place editing operation has been confirmed, or **false** otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaisePasted Method

Raises the appropriate Pasted event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaisePasted (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Sub RaisePasted( _
    item As DiagramItem _
)
```

Parameters

item
The DiagramItem related to the event.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseSelecting Method

Raises the appropriate Selecting event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RaiseSelecting (
    DiagramItem item,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public Function RaiseSelecting( _
    item As DiagramItem, _
    mousePosition As Point _
) As Boolean
```

Parameters

item

The DiagramItem related to the event.

mousePosition
The current mouse position, in document coordinates.

Return Value

true if the operation has been confirmed; otherwise, **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseTreeItemTextEditing Method

Raises the TreeItemTextEditing event to validate entering inplace edit mode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RaiseTreeItemTextEditing (
    TreeViewItem item
)
```

Visual Basic Copy Code

```
Public Function RaiseTreeItemTextEditing( _
    item As TreeViewItem _
) As Boolean
```

Parameters

item
The TreeViewItem instance related to the event.

Return Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RaiseViewRemoved Method

Raises the ViewRemoved event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaiseViewRemoved ()
```

Visual Basic Copy Code

```
Public Sub RaiseViewRemoved ()
```

See Also

Diagram.RegisterClass Method

Overload List

Name	Description
Diagram.RegisterClass (Type, String, Int32)	Registers a diagram item or lane header class for serialization support.
Diagram.RegisterClass (Type, String, Int32, String)	Registers a diagram item or lane header class for serialization support.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RegisterClass Method (Type, String, Int32)

Registers a diagram item or lane header class for serialization support.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static void RegisterClass (
    Type classType,
    string classId,
    int classVersion
)
```

Visual Basic Copy Code

```
Public Shared Sub RegisterClass( _
    classType As Type, _
    classId As String, _
    classVersion As Integer _
)
```

Parameters

classType
A Type instance identifying the class.

classId
A class identifier to use when saving and loading objects of the specified type.

classVersion
A revision number of the serialization format.

See Also

RegisterClass Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RegisterClass Method (Type, String, Int32, String)

Registers a diagram item or lane header class for serialization support.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static void RegisterClass (
    Type classType,
    string classId,
    int classVersion,
    string clientClass
)
```

Visual Basic Copy Code

```
Public Shared Sub RegisterClass( _
    classType As Type, _
    classId As String, _
    classVersion As Integer, _
    clientClass As String _
)
```

Parameters

classType
A Type instance identifying the class.

classId
A class identifier to use when saving and loading objects of the specified type.

classVersion
A revision number of the serialization format.

clientClass
A class identifier to use when serializing objects of the specified type to JSON.

See Also

RegisterClass Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RegisterDefaultStyle Method

Registers a default style for the objects of the specified type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static void RegisterDefaultStyle (
    Type type,
    Style style
)
```

Visual Basic Copy Code

```
Public Shared Sub RegisterDefaultStyle( _
    type As Type, _
    style As Style _
)
```

Parameters

type
A **System.Type** specifying the type of diagram items to which the style applies.

style
The Style to apply to the specified type of items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RegisterItemClass Method

Overload List

Name	Description
Diagram.RegisterItemClass (Type, String, Int32)	Registers a diagram item class for serialization support.
Diagram.RegisterItemClass (Type, String, Int32, String)	Registers a diagram item class for serialization support.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RegisterItemClass Method (Type, String, Int32)

Registers a diagram item class for serialization support.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static void RegisterItemClass (
    Type itemClass,
    string classId,
    int classVersion
)
```

Visual Basic Copy Code

```
Public Shared Sub RegisterItemClass( _
    itemClass As Type, _
    classId As String, _
    classVersion As Integer _
)
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when saving and loading items of the specified type.

classVersion
A revision number of the item's class serialization format.

Remarks

When using custom item types, use this method to enable saving and loading the custom item instances to/from XML or binary files. The class id string is written before each item's data, and when loading items later that string indicates what class instance to create and let it load the subsequent data. The following class identifiers are reserved and should not be used by your application: "std:DiagramLink", "std:ShapeNode", "std:TableNode", "std:ContainerNode" and "std:ControlNode".

The version identifier must be increased when the format in which items are saved changes. It can be used when loading files to let your application load items data from an older format.

Note

All custom item classes must provide a parameterless constructor, or one that takes a Diagram argument. Failing to do so would result in exceptions during the load process, since the diagram will be unable to create items from the custom type.

See Also

RegisterItemClass Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RegisterItemClass Method (Type, String, Int32, String)

Registers a diagram item class for serialization support.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static void RegisterItemClass (
    Type itemClass,
    string classId,
    int classVersion,
    string clientClass
)
```

Visual Basic Copy Code

```
Public Shared Sub RegisterItemClass( _
    itemClass As Type, _
    classId As String, _
    classVersion As Integer, _
    clientClass As String _
)
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when saving and loading items of the specified type.

classVersion
A revision number of the item's class serialization format.

clientClass
A class identifier to use when serializing item of the specified type to JSON.

See Also

RegisterItemClass Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RemoveItem Method

Invoked by the AddItemCmd and RemoveItemCmd classes to effectively remove the specified item from the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void RemoveItem (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Protected Sub RemoveItem( _
    item As DiagramItem _
)
```

Parameters

item
The item being removed from the diagram.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AddItem Method

Diagram.ResetAlignmentGuides Method

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ResetAlignmentGuides ()
```

Visual Basic Copy Code

```
Public Sub ResetAlignmentGuides ()
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ResizeToFitItem Method

Resizes the document extents so that the specified item is contained within.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ResizeToFitItem (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Sub ResizeToFitItem( _
    item As DiagramItem _
)
```

Parameters

item
The diagram item contained in the document area.

Remarks

Resizes the document area to fit the specified diagram item.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ResizeToFitItems Method
AutoResize Property

Diagram.ResizeToFitItems Method

Resizes the document scrollable area so it fits all diagram items.

Overload List

Name	Description
Diagram.ResizeToFitItems (Double)	Resizes the document scrollable area so it fits all diagram items.
Diagram.ResizeToFitItems (Double, Boolean)	Resizes the document scrollable area so it fits all diagram items, optionally considering only the visible items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AutoResize Property
Bounds Property

Diagram.ResizeToFitItems Method (Double)

Resizes the document scrollable area so it fits all diagram items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ResizeToFitItems (
    double borderGap
)
```

Visual Basic Copy Code

```
Public Sub ResizeToFitItems( _
    borderGap As Double _
)
```

Parameters

borderGap
A gap that should be left as a border around the diagram.

Remarks

Resizes the document area to be just as big as needed, for example, the smallest rectangle containing all diagram items. For aesthetic reasons, additional space can be left around the outmost items as a margin. Invoking the method on an empty document produces no result.

See Also

ResizeToFitItems Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AutoResize Method
Bounds Property

Diagram.ResizeToFitItems Method (Double, Boolean)

Resizes the document scrollable area so it fits all diagram items, optionally considering only the visible items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ResizeToFitItems (
    double borderGap,
    bool onlyVisible
)
```

Visual Basic Copy Code

```
Public Sub ResizeToFitItems( _
    borderGap As Double, _
    onlyVisible As Boolean _
)
```

Parameters

borderGap
A gap that should be left as a border around the diagram.

onlyVisible
Specifies whether only the bounds of the Visible items should be included in the new document area.

Remarks

Resizes the document area to be just as big as needed, for example the smallest rectangle containing all diagram items. For aesthetic reasons, additional space can be left around the outmost items as a margin. Invoking the method on an empty document produces no result.

See Also

ResizeToFitItems Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AutoResize Method
Bounds Property

Diagram.ResumeLinkRouter Method

Called when the control ends a batch update operation that might affect many links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ResumeLinkRouter (
    bool routeDeferredLinks
)
```

Visual Basic Copy Code

```
Public Sub ResumeLinkRouter( _
    routeDeferredLinks As Boolean _
)
```

Parameters

routeDeferredLinks
true if the links should be routed, otherwise false.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RouteAllLinks Method

Overload List

Name	Description
Diagram.RouteAllLinks ()	Routes all links in the diagram in a way they do not cross nodes.
Diagram.RouteAllLinks (DiagramLinkCollection)	Reroutes the specified links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RouteAllLinks Method ()

Routes all links in the diagram in a way they do not cross nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RouteAllLinks ()
```

Visual Basic Copy Code

```
Public Sub RouteAllLinks ()
```

Remarks

Each link is routed so that it goes the shortest possible way between the nodes it links without crossing any other nodes. To do so, this method might modify the number of link segments. The value of AutoRoute is not considered when **RouteAllLinks** executes.

RoutingOptions allows tuning the routing algorithm. The most important options are the cost values exposed as LengthCost, TurnCost and CrossingCost properties. They greatly influence the shape of the generated route.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
RouteLinks Property
LinkRouter Property

Diagram.RouteAllLinks Method (DiagramLinkCollection)

Reroutes the specified links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RouteAllLinks (
    DiagramLinkCollection linksToRoute
)
```

Visual Basic Copy Code

```
Public Sub RouteAllLinks( _
    linksToRoute As DiagramLinkCollection _
)
```

Parameters

linksToRoute
A list of diagram links.

See Also

RouteAllLinks Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SaveSelectionToJson Method

Saves the diagram selection to a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string SaveSelectionToJson ()
```

Visual Basic Copy Code

```
Public Function SaveSelectionToJson () As String
```

Return Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SaveToJson Method

Saves the diagram to a JSON-formatted string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual string SaveToJson ()
```

Visual Basic Copy Code

```
Public Overridable Function SaveToJson () As String
```

Return Value

A string containing the diagram's JSON representation.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LoadFromJson Method
SaveToJsonFile Method

Diagram.SaveToJsonAsync Method

Saves the diagram to a JSON-formatted string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task<string> SaveToJsonAsync ()
```

Visual Basic Copy Code

```
Public Function SaveToJsonAsync () As Task(Of String)
```

Return Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SaveToJsonFile Method

Saves the diagram into a JSON file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToJsonFile (
    string fileName
)
```

Visual Basic Copy Code

```
Public Sub SaveToJsonFile( _
    fileName As String _
)
```

Parameters

fileName
A string specifying the file name or full path.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LoadFromJsonFile Method
SaveToJson Method

Diagram.SaveToJsonFileAsync Method

Saves the diagram into a JSON file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task SaveToJsonFileAsync (
    string fileName
)
```

Visual Basic Copy Code

```
Public Function SaveToJsonFileAsync( _
    fileName As String _
) As Task
```

Parameters

fileName

Return Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SaveToJsonObject Method

Saves the diagram to a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void SaveToJsonObject (
    JsonObject diagramObj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Public Overridable Sub SaveToJsonObject( _
    diagramObj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

diagramObj
A JsonObject the diagram's data should be stored into.

context
A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SaveToString Method

Encodes the diagram contents into an ASCII string using the specified format and optionally clearing the Dirty flag.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string SaveToString (
    SaveToStringFormat format,
    bool clearDirty
)
```

Visual Basic Copy Code

```
Public Function SaveToString( _
    format As SaveToStringFormat, _
    clearDirty As Boolean _
) As String
```

Parameters

format
A member of the SaveToStringFormat enumeration.

clearDirty
Indicates whether the Dirty flag should be cleared.

Return Value

A string containing the encoded diagram data.

Remarks

The Base16 and **Base64** formats represent the data returned by SaveToStream as an ASCII string. There are no special characters in the returned string, so the result can be easily stored in a database field or embedded into an XML document. That might be used also for transferring a diagram between a web page with embedded MvcDiagram and a web server, passing the string as a value of a hidden HTML field.

When using **Xml** or **CompressedXml** as the first argument, the diagram is saved in XML format. The same format is supported by the SaveToXml and LoadFromXml methods. **CompressedXml** specifies that the XML document should be further compressed in ZIP format and base-64 encoded.

Exceptions

Exception type	Condition
NotImplementedException	When saving the diagram using the CompressedXml flag under .NET Framework 1.1 or earlier.

See Also

SaveToString Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LoadFromString Method
SaveToFile Method

Diagram.SaveToXml Method

Saves the diagram to an XML document.

Overload List

Name	Description
Diagram.SaveToXml (Stream)	Saves the diagram to the specified stream in XML format.
Diagram.SaveToXml (String)	Saves the diagram to the specified file as an XML document.
Diagram.SaveToXml (XmlDocument)	Saves the diagram to the specified XML document.
Diagram.SaveToXml (XmlDocument, Boolean)	Saves the diagram to the specified XML document.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
SaveToFile Method

Diagram.SaveToXml Method (Stream)

Saves the diagram to the specified stream in XML format.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToXml (
    Stream stream
)
```

Visual Basic

```
Public Sub SaveToXml( _
    stream As Stream _
)
```

Parameters

stream

See Also

SaveToXml Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SaveToXml Method (String)

Saves the diagram to the specified file as an XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void SaveToXml (
    string fileName
)
```

Visual Basic

```
Public Sub SaveToXml( _
    fileName As String _
)
```

Parameters

fileName
The full path and name of the disk file where the diagram should be saved.

Remarks

The first overloaded **SaveToXml** method works by creating an **XmlDocument** instance, calling the second overloaded method, and writing the document content to a file.

Diagram data can be loaded from an XML file or document using the LoadFromXml method.

See Also

SaveToXml Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
SaveToFile Method

Diagram.SaveToXml Method (XmlDocument)

Saves the diagram to the specified XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public virtual void SaveToXml (
    XmlDocument document
)
```

Visual Basic Copy Code

```
Public Overridable Sub SaveToXml( _
    document As XmlDocument _
)
```

Parameters

document
A .NET **XmlDocument** object where the diagram should be saved.

Remarks

Diagram data can be loaded from an XML file or document using the LoadFromXml method.

See Also

SaveToXml Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
SaveToFile Method

Diagram.SaveToXml Method (XmlDocument, Boolean)

Saves the diagram to the specified XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void SaveToXml (
    XmlDocument document,
    bool includeUnalteredProperties
)
```

Visual Basic Copy Code

```
Public Overridable Sub SaveToXml( _
    document As XmlDocument, _
    includeUnalteredProperties As Boolean _
)
```

Parameters

document
A .NET **XmlDocument** object where the diagram should be saved.

includeUnalteredProperties
true if properties that have their default value should be saved; otherwise, **false**.

Remarks

Diagram data can be loaded from an XML file or document using the LoadFromXml method.

See Also

SaveToXml Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SaveToXmlAsync Method

Overload List

Name	Description
Diagram.SaveToXmlAsync (String)	Saves the diagram into an XML file.
Diagram.SaveToXmlAsync (XmlDocument)	Saves the diagram into an XML document.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SaveToXmlAsync Method (String)

Saves the diagram into an XML file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task SaveToXmlAsync (
    string fileName
)
```

Visual Basic Copy Code

```
Public Function SaveToXmlAsync( _
    fileName As String _
) As Task
```

Parameters

fileName

Return Value

See Also

SaveToXmlAsync Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SaveToXmlAsync Method (XmlDocument)

Saves the diagram into an XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task SaveToXmlAsync (
    XmlDocument document
)
```

Visual Basic

```
Public Function SaveToXmlAsync( _
    document As XmlDocument _
) As Task
```

Parameters

document

Return Value

See Also

SaveToXmlAsync Method Overload List
Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SetShadowOffset Method

Sets the default offset at which items shadows are drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void SetShadowOffset (
    double x,
    double y
)
```

Visual Basic

```
Public Sub SetShadowOffset( _
    x As Double, _
    y As Double _
)
```

Parameters

x
The horizontal offset of the item shadow.

y
The vertical offset of the item shadow.

Remarks

Use the method to draw shadows that are closer or further to the item they belong to. The values are applied respectively to the ShadowOffsetX and ShadowOffsetY properties.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShadowOffsetX Property
ShadowOffsetY Property
ShadowBrush

Diagram.SuspendLinkRouter Method

Called when the control starts a batch update operation that might affect many links. The diagram collects links specified through Route and RouteAllLinks calls and defers finding their paths to when ResumeLinkRouter is called.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SuspendLinkRouter ()
```

Visual Basic Copy Code

```
Public Sub SuspendLinkRouter ()
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.UpdateMeasureTextScale Method

Gets or sets the scale to use when measuring texts. This should be effectively set to the value of DiagramView.Zoom / 100.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void UpdateMeasureTextScale (
    float measureTextScale
)
```

Visual Basic Copy Code

```
Public Sub UpdateMeasureTextScale( _
    measureTextScale As Single _
)
```

Parameters

measureTextScale

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.UpdateRuntimeIndices Method

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void UpdateRuntimeIndices ()
```

Visual Basic Copy Code

```
```

```
Public Sub UpdateRuntimeIndices ()
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.UpdateZOrder Method

Immediately sorts the Z-order list if Z-order has been invalidated. Otherwise it will be sorted next time the diagram draws its contents.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void UpdateZOrder ()
```

Visual Basic Copy Code

```
Public Sub UpdateZOrder ()
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram Properties

Diagram.ActiveItem Property

Gets or sets the active diagram element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem ActiveItem { get; set; }
```

Visual Basic Copy Code

```
Public Property ActiveItem As DiagramItem
```

Property Value

A reference to the DiagramItem instance that is currently active.

Remarks

Use this property to get the active item - the one selected by the user or the last put in a multiple Selection. It is shown with white selection handles by default.

See Also

Diagram Members
Diagram Class

Diagram.ActiveItemHandlesStyle Property

Gets the visual style of adjustment handles of the active item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HandlesVisualStyle ActiveItemHandlesStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ActiveItemHandlesStyle As HandlesVisualStyle
```

Property Value

An instance of the HandlesVisualStyle class.

Remarks

This style object provides Pen and Brush attributes that let you customize the appearance of adjustment handles drawn around the active item.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ActiveItem Property

Diagram.ActiveLayer Property

Gets or sets the zero-based index of the active layer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int ActiveLayer { get; set; }
```

Visual Basic Copy Code

```
Public Property ActiveLayer As Integer
```

Property Value

An integer value that specifies the index of the active layer.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.AdjustmentHandlesSize Property

Gets or sets the size of the handles, which appear around selected objects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double AdjustmentHandlesSize { get; set; }
```

Visual Basic Copy Code

```
Public Property AdjustmentHandlesSize As Double
```

Property Value

A **double** value specifying the height of the selection handles. The default is 2.

Remarks

Use the property to set the size of the manipulation handles, which appear around selected items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ModificationStart Property
SelectedItemHandlesColor Property

Diagram.AlignmentGuidePen Property

Gets or sets the pen used to draw alignment guides.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen AlignmentGuidePen { get; set; }
```

Visual Basic Copy Code

```
Public Property AlignmentGuidePen As Pen
```

Property Value

An instance of the Pen class.

Remarks

This pen is used to draw alignment guides shown when AutoAlignNodes is enabled.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AutoAlignDistance Property

Diagram.AlignToGrid Property

Gets or sets a value indicating whether the diagram items is aligned to a grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AlignToGrid { get; set; }
```

Visual Basic Copy Code

```
Public Property AlignToGrid As Boolean
```

Property Value

true if the grid is activated; otherwise, **false**. The default is **true**.

Remarks

When the property is set to **true**, the sides of every newly created diagram item are aligned to the nearest grid points.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShowGrid Property
GridColor Property
GridSizeX Property
GridSizeY Property
AlignPointToGrid Method
AutoAlignNodes Property

Diagram.AllowLinksRepeat Property

Gets or sets a value indicating whether more than one link can connect the same origin and destination diagram nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowLinksRepeat { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowLinksRepeat As Boolean
```

Property Value

true if repeated links are allowed; otherwise, **false**. The default is **true**.

Remarks

Specifies whether more than one link can connect the same origin and destination diagram nodes. If set to **false**, two diagram nodes can be connected with at most one link in a single direction.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AllowSelfLoops Property

Diagram.AllowMultipleResize Property

Gets or sets whether multiple selected nodes can be resized simultaneously.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowMultipleResize { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowMultipleResize As Boolean
```

Property Value

true to let the user resize multiple selected nodes; otherwise, **false**. The default value is **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.AllowSelfLoops Property

Gets or sets a value indicating whether self-loops are allowed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowSelfLoops { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowSelfLoops As Boolean
```

Property Value

true if self loops are allowed; otherwise, **false**. The default is **true**.

Remarks

Use the property to specify whether the user is allowed to draw self loops - links that connect a node to itself.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AllowLinksRepeat Property

Diagram.AllowSplitLinks Property

Gets or sets a value indicating whether link segments can be added and removed interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowSplitLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowSplitLinks As Boolean
```

Property Value

true if interactive adding/removing of link segments allowed; **false** if not.

Remarks

You can insert new link segments interactively by clicking and dragging a point of a selected link's segment. For polyline-style links this creates one additional segment by splitting the existing one into two parts. For cascading links there are two segments inserted, keeping intact the alternating pattern of horizontal and vertical segments, each orthogonal to its neighbors.

Segments can be merged by moving their joining point until the angle between reaches certain threshold value. In case of cascading and Bezier links, segments can be merged by placing adjacent control points close to each other, as set in MergeThreshold.

This property also affects how users can change the orientation of cascading links. For more details see CascadeOrientation.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LinkSegments Property
LinkShape Property

Diagram.AllowUnanchoredLinks Property

Gets or sets a value indicating whether links can be connected to nodes to which there aren't anchor points assigned.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowUnanchoredLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowUnanchoredLinks As Boolean
```

Property Value

true to let links connect to nodes without anchor points; otherwise, **false**. The default is **true**.

Remarks

AllowUnanchoredLinks applies to nodes whose **AnchorPattern** is not set or does not contain incoming or outgoing anchor points available. If the property is enabled, users can connect links to any point of such nodes. If disabled, users cannot connect links to such nodes.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AnchorPoint.AllowIncoming Property
AnchorPoint.AllowOutgoing Property

Diagram.AllowUnconnectedLinks Property

Gets or sets a value indicating whether users are allowed to draw links that are not connected to any node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowUnconnectedLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowUnconnectedLinks As Boolean
```

Property Value

true to allow drawing unconnected links; otherwise, **false**. The default is **false**.

Remarks

If the property is set to **true**, users will be able to draw links from or to any point of the document. Depending on which end of a link is unconnected, its Origin or Destination nodes might be of type DummyNode.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AllowUnanchoredLinks Property

Diagram.AutoAlignDistance Property

Gets or sets the maximal horizontal or vertical distance at which a node aligns to the borders of other nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double AutoAlignDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoAlignDistance As Double
```

Property Value

A **double** value specifying the maximal alignment distance. The default value is 3.

Remarks

To enable node alignment, set AutoAlignNodes to true. When either the horizontal or vertical borders of the dragged node are within the specified distance from the respective borders of a nearby node, the dragged node snaps to a position where the borders are aligned.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AlignmentGuidePen Property

Diagram.AutoAlignNodes Property

Gets or sets whether nodes should automatically align to the borders of other nodes during user interaction.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoAlignNodes { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoAlignNodes As Boolean
```

Property Value

true to align nodes, otherwise **false**. The default value is **false**.

Remarks

When either the horizontal or vertical borders of the dragged node are within AutoAlignDistance from the respective borders of a nearby node, the dragged node snaps to a position where the borders are aligned. In addition, an alignment guide that indicates which borders are aligned is drawn using the pen specified through AlignmentGuidePen.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AlignToGrid Property
AutoSnapLinks Property
QueryAlignTarget Event

Diagram.AutoArrangeAvoidContainers Property

Gets or sets a value indicating whether auto-arranged link labels should avoid containers.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoArrangeAvoidContainers { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoArrangeAvoidContainers As Boolean
```

Property Value

true to avoid containers, or **false** otherwise. The default value is **true**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.AutoArrangeAvoidSegments Property

Gets or sets a value indicating whether auto-arranged link labels should avoid link segments.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoArrangeAvoidSegments { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoArrangeAvoidSegments As Boolean
```

Property Value

true to avoid link segments, or **false** otherwise. The default value is **true**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.AutoCloseDistance Property

Gets or sets the maximum distance between first and last points of a FreeFormNode for which the node's outline is closed automatically.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Nullable<Double> AutoCloseDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoCloseDistance As Nullable(Of Double)
```

Property Value

An optional **double** value specifying upper limit of distance between end points where the form should close automatically. The default **null** value specifies that the form should always close.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.AutoDeleteLinks Property

Gets or sets whether links are deleted automatically when their Origin or Destination is deleted.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoDeleteLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoDeleteLinks As Boolean
```

Property Value

true to delete links along with the nodes they connect, or **false** otherwise. The default value is **true**.

Remarks

If this property is set to **false**, deleting a node will leave its links in the diagram as unconnected links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.AutoHighlightRows Property

Gets or sets whether a table row should be highlighted when the user clicks on it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoHighlightRows { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoHighlightRows As Boolean
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.AutoResize Property

Gets or sets a value indicating whether the document area must be resized automatically so it fits the diagram contents.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AutoResize AutoResize { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoResize As AutoResize
```

Property Value

A member of the AutoResize enumeration. The default is **AutoResize.RightAndDown**.

Remarks

If automatic resizing is enabled, the size of the document scrollable area is updated each time an item is created, deleted or moved, either by end-users or by calling of methods. The following AutoResize values specify in what directions the diagram area is allowed to grow:

Member name	Description
None	Do not enlarge the diagram area automatically.
RightAndDown	Enlarge the diagram area to the right and down if an item is moved outside the boundaries in these directions.
AllDirections	Enlarge the diagram area in any direction if an item is moved outside the boundaries.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Bounds Property
ResizeToFitItems Method

Diagram.AutoSnapDistance Property

Specifies the maximum distance at which links auto-snap to nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double AutoSnapDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoSnapDistance As Double
```

Property Value

A **double** value specifying the maximum auto-snap distance. The default value is 16.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AutoSnapLinks Property

Diagram.AutoSnapLinks Property

Gets or sets the default value for the AutoSnapToNode property of new links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoSnapLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoSnapLinks As Boolean
```

Property Value

true to snap links to nearest node, otherwise **false**; the default value is **false**.

Remarks

Only nodes at a distance smaller than AutoSnapDistance are considered. A link does not snap to its current origin or destination nodes if that would create a self-loop.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DynamicLinks Property
AutoAlignNodes Property

Diagram.BackBrush Property

Gets or sets the brush used for painting the diagram background.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush BackBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property BackBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class. The default is a solid brush initialized with Color.FromArgb(170, 170, 200).

Remarks

This property enables more spectacular background painting with effects such as gradients, textures and color patterns.

Example

The following sample paints the diagram background with a gradient brush:

C# Copy Code

```
private void Form1_Load(object sender, System.EventArgs e)
{
    diagram.BackBrush = new MindFusion.Drawing.LinearGradientBrush(
        Color.LightBlue, Color.LightCyan, 30);
}
```

Visual Basic Copy Code

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

    diagram.BackBrush = New MindFusion.Drawing.LinearGradientBrush( _
        Color.LightBlue, Color.LightCyan, 30)

End Sub
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Brush and Pen Classes

Diagram.BackgroundImage Property

Gets or sets the image displayed as background of this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage BackgroundImage { get; set; }
```

Visual Basic Copy Code

```
Public Property BackgroundImage As IImage
```

Property Value

A **Microsoft.Maui.Graphics.IImage** instance.

Remarks

This image is painted as a background of the diagram, behind the alignment grid and all items. Its position is controlled by the BackgroundImageAlign property. To have the background image appear on printed pages, set the EnableBackgroundImage property of **DiagramView.PrintOptions** to **true**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
BackgroundImageAlign Property
BackBrush Property

Diagram.BackgroundImageAlign Property

Gets or sets a value indicating how the background image is positioned, relatively to the scrollable document area.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ImageAlign BackgroundImageAlign { get; set; }
```

Visual Basic Copy Code

```
Public Property BackgroundImageAlign As ImageAlign
```

Property Value

A member of the ImageAlign enumeration. The default is **ImageAlign.TopLeft**.

Remarks

The background image can be aligned to a corner, stretched, centered, or tiled to cover the control's area.

Example

To have a background image stretched along the drawing surface, use code like the following:

C# Copy Code

```
diagram.BackgroundImage = Image.FromFile("back.jpg");
diagram.BackgroundImageAlign = ImageAlign.Stretch;
```

Visual Basic Copy Code

```
diagram.BackgroundImage = Image.FromFile("back.jpg")
Diagram.BackgroundImageAlign = ImageAlign.Stretch
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ImageAlign Enumeration
BackgroundImage Property

Diagram.BackgroundImageUrl Property

URL of the image painted as diagram's background.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string BackgroundImageUrl { get; set; }
```

Visual Basic Copy Code

```
Public Property BackgroundImageUrl As String
```

Property Value

A string specifying the location of the image file.

Remarks

This property is used in the WebForms version of the control. Its advantage is that it allows for the diagram items to be loaded faster on the client side, while the background image is loaded asynchronously.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.BarcodeNodeStyle Property

Gets the theme-level style for barcode nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public BarcodeNodeStyle BarcodeNodeStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property BarcodeNodeStyle As BarcodeNodeStyle
```

Property Value

A **BarcodeNodeStyle** instance that defines the default appearance of **BarcodeNode**s.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.Bounds Property

Gets or sets a value representing the size of the diagram's scrollable area.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect Bounds { get; set; }
```

Visual Basic Copy Code

```
Public Property Bounds As Rect
```

Property Value

An instance of the **Microsoft.Maui.Graphics.Rect** structure. The default size of the document is the standard A4 size (210x297 mm).

Remarks

Use this property to specify how large a document should be, that is, how far it can be scrolled. If **ShowScrollbars** is enabled, setting the document extents also updates the scrollbar ranges. After creating a diagram programmatically, you might wish to call ResizeToFitItems to make the scrollable area just big enough to contain all diagram elements.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
MeasureUnit Property
RestrictItemsToBounds Property
AutoResize Property

Diagram.BoundsPen Property

The pen with which the document bounds should be drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen BoundsPen { get; set; }
```

Visual Basic Copy Code

```
Public Property BoundsPen As Pen
```

Property Value

A Pen instance. The default value is null.

Remarks

This pen is used to draw borders around the diagram bounding rectangle, which could be useful when the **DiagramView** is larger than the diagram.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ExteriorBrush Property

Diagram.BranchIndicatorColor Property

Gets or sets the color of link branch indicators.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Color BranchIndicatorColor { get; set; }
```

Visual Basic Copy Code

```
Public Property BranchIndicatorColor As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** instance specifying the color of branch indicators. The default value is **LightGreen**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
BranchIndicatorSize Property
LinkBranchIndicator Property

Diagram.BranchIndicatorSize Property

Gets or sets the size of link branch indicators.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double BranchIndicatorSize { get; set; }
```

Visual Basic Copy Code

```
Public Property BranchIndicatorSize As Double
```

Property Value

A **double** value specifying the size of branch indicators. The default value is 4.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LinkBranchIndicator Property
BranchIndicatorColor Property

Diagram.CellCustomDraw Property

Gets or sets the default type of custom painting done on the cells of tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CustomDraw CellCustomDraw { get; set; }
```

Visual Basic Copy Code

```
Public Property CellCustomDraw As CustomDraw
```

Property Value

A member of the CustomDraw enumeration. The default is **CustomDraw.None**.

Remarks

The CellCustomDraw member of new tables is initialized from this property.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
TableCustomDraw Property

Diagram.CellFrameStyle Property

Gets or sets the default cell border style of new tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CellFrameStyle CellFrameStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property CellFrameStyle As CellFrameStyle
```

Property Value

A member of the CellFrameStyle enumeration. The default is **CellFrameStyle.System3D**.

Remarks

Use this property to specify the default border style of cells in new tables. See CellFrameStyle enumeration for more details.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.CompositeNodeStyle Property

Gets the theme-level style for composite nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CompositeNodeStyle CompositeNodeStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property CompositeNodeStyle As CompositeNodeStyle
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ContainerCaption Property

Gets or sets the default value for the Caption property of containers.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ContainerCaption { get; set; }
```

Visual Basic Copy Code

```
Public Property ContainerCaption As String
```

Property Value

A string specifying the initial caption text of containers. The default is "Container".

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ContainerCaptionHeight Property

Gets or sets the default value for the CaptionHeight property of containers.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ContainerCaptionHeight { get; set; }
```

Visual Basic Copy Code

```
Public Property ContainerCaptionHeight As Double
```

Property Value

A **double** value specifying the initial height of containers' caption bars. The default is 5.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ContainerHighlightPen Property

Gets or sets the default value for the HighlightPen property of containers.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen ContainerHighlightPen { get; set; }
```

Visual Basic Copy Code

```
Public Property ContainerHighlightPen As Pen
```

Property Value

A Pen instance specifying the initial highlight pen of containers. The default is a pen with **Color.LightSkyBlue** color and width 0.5.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ContainerMargin Property

Gets or sets the default value for the Margin property of containers.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ContainerMargin { get; set; }
```

Visual Basic Copy Code

```
Public Property ContainerMargin As Double
```

Property Value

A **double** value specifying the initial margin size of containers. The default is 10.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ContainerMinimumSize Property

Gets or sets the default value for the MinimumSize property of containers.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size ContainerMinimumSize { get; set; }
```

Visual Basic Copy Code

```
Public Property ContainerMinimumSize As Size
```

Property Value

A **Size** value specifying the initial minimum size of containers. The default is 50x40.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ContainerNodeStyle Property

Gets the theme-level style for container nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNodeStyle ContainerNodeStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ContainerNodeStyle As ContainerNodeStyle
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ContainersFoldable Property

Gets or sets the default value for the Foldable property of containers.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ContainersFoldable { get; set; }
```

Visual Basic

```
Public Property ContainersFoldable As Boolean
```

Property Value

A boolean value specifying whether containers are initially foldable. The default is **true**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.CrossingRadius Property

Gets or sets the radius length of decorations displayed at link intersection points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double CrossingRadius { get; set; }
```

Visual Basic

```
Public Property CrossingRadius As Double
```

Property Value

A **double** value specifying the radius of the arcs drawn at link intersection points.

Remarks

If LinkCrossings is set to **Cut**, link segments are cut at intersection points and this property defines the length of the break left between adjacent cut points. If **LinkCrossings** is set to **Arcs**, this property defines the radius of the arcs drawn over intersection points.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
RoundedLinksRadius Property
DiagramLink.CrossingRadius Property

Diagram.DefaultShape Property

Gets or sets the default shape of shape nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Shape DefaultShape { get; set; }
```

Visual Basic

```
Public Property DefaultShape As Shape
```

Property Value

An instance of the Shape class. The default is the RoundRect shape.

Remarks

The Shape property of newly created shape nodes is initialized with the value of this property. It defines a series of lines, arcs and curves composed in a complex shape. There are 87 **predefined shapes** exposed as static members of the Shape class. You can also define your own shapes as shown below.

Example

The following example demonstrates how to create a custom shape and set it as a default in the diagram:

C# Copy Code

```
private void example_Click(object sender, System.EventArgs e)
{
    // Define a custom shape
    // this is the same as: fc.DefaultShape = Shape.FromId("Tape");
    diagram.DefaultShape = new Shape(
        new ElementTemplate[] {
            new LineTemplate(50, 100, 85, 100),
            new LineTemplate(85, 100, 85, 90),
            new LineTemplate(85, 90, 70, 90),
            new BezierTemplate(70, 90, 85, 90, 100, 75, 100, 50),
            new BezierTemplate(100, 50, 100, 25, 75, 0, 50, 0),
            new BezierTemplate(50, 0, 25, 0, 0, 25, 0, 50),
            new BezierTemplate(0, 50, 0, 75, 25, 100, 50, 100) },
        FillMode.Winding);
}
```

Visual Basic Copy Code

```
Private Sub example_Click(ByVal sender As Object, ByVal e As System.EventArgs)

    ' Define a custom shape
    ' this is the same as: fc.DefaultShape = Shape.FromId("Tape")
    Diagram.DefaultShape = New Shape( _
        New ElementTemplate() { _
            New LineTemplate(50, 100, 85, 100), _
            New LineTemplate(85, 100, 85, 90), _
            New LineTemplate(85, 90, 70, 90), _
            New BezierTemplate(70, 90, 85, 90, 100, 75, 100, 50), _
            New BezierTemplate(100, 50, 100, 25, 75, 0, 50, 0), _
            New BezierTemplate(50, 0, 25, 0, 0, 25, 0, 50), _
            New BezierTemplate(0, 50, 0, 75, 25, 100, 50, 100)}, _
        FillMode.Winding)

End Sub
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShapeNode.Shape Property
Shape.FromId Method
Shape.Shapes Property
ShapeOrientation Property

Diagram.DiagramLinkStyle Property

Gets the theme-level style for diagram links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkStyle DiagramLinkStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property DiagramLinkStyle As DiagramLinkStyle
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.DiagramStyle Property

Gets the theme-level style for the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramStyle DiagramStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property DiagramStyle As DiagramStyle
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.Dirty Property

Gets or sets a value indicating whether the diagram has changed since the last save.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Dirty { get; set; }
```

Visual Basic Copy Code

```
Public Property Dirty As Boolean
```

Property Value

A boolean value that indicates whether the diagram has changed.

Remarks

MvcDiagram tracks all changes that are done on the diagram contents, either programmatically or by users. If there are changes that are still not saved, the property has value of **true**. The **Dirty** flag is cleared automatically when a document is loaded from file or stream. When a document is saved, the **Dirty** flag can be cleared as specified by an argument of the Save* methods.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DirtyChanged Event
SaveToStream Method
LoadFromStream Method
SaveToFile Method
LoadFromFile Method

Diagram.DisabledHandlesStyle Property

Gets the visual style of disabled adjustment handles.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HandlesVisualStyle DisabledHandlesStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property DisabledHandlesStyle As HandlesVisualStyle
```

Property Value

An instance of the HandlesVisualStyle class.

Remarks

This style object provides Pen and Brush attributes that let you customize the appearance of disabled adjustment handles.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramNode.EnabledHandles Property

Diagram.DynamicLinks Property

Gets or sets the default value for the Dynamic property of new links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool DynamicLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property DynamicLinks As Boolean
```

Property Value

true if dynamic links are enabled; otherwise, **false**. The default is **false**.

Remarks

Use the property to specify whether links automatically reposition their end segments, so they keep pointing to the centers of connected nodes. If enabled, moving a connected node recalculates the position of link's end point, placing it at appropriate

location of the node outline. If the node has anchor points assigned to it, the end point is attached to one of them.

If the value is **false**, links do not look for the center of the origin and destination node, but keep their relative position as set when they were created - from the left corner of the first node to the top of the other node for instance.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.Dynamic Property
AllowSplitLinks Property

Diagram.EffectiveBackBrush Property

Gets the currently effective background brush of the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush EffectiveBackBrush { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectiveBackBrush As Brush
```

Property Value

Remarks

This property returns Brush if it is not null, or the brush from the associated style or theme otherwise.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.EffectsProvider Property

Gets or sets an object used to generate visual effects for shape nodes such as embossment and shadow dispersion.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IEffectsProvider EffectsProvider { get; set; }
```

Visual Basic Copy Code

```
Public Property EffectsProvider As IEffectsProvider
```

Property Value

An object that implements the IEffectsProvider interface. The default is **null** (**Nothing** in Visual Basic).

Remarks

For performance reasons, embossment and shadow dispersion effects are implemented through code that processes large amounts of image data by directly accessing the memory allocated to the image. The .NET runtime considers such code unsafe, and the default CAS settings prevent assemblies that contain unsafe code from being loaded in Internet Explorer or IIS. In order

to keep the MvcDiagram CAS requirements low, the unsafe code resides in a separate assembly - *MindFusion.Diagramming.EffectsProvider.dll*. If you wish to use the EnableEmbossEffects and EnableShadowEffects properties, you must add to your project a reference to that assembly, instantiate an EffectsProvider object, and assign it to the **EffectsProvider** property.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.EnableEmbossEffects Property

Gets or sets a value indicating whether emboss effect for shape nodes is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableEmbossEffects { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableEmbossEffects As Boolean
```

Property Value

true to enable embossment of shape nodes; otherwise, **false**. The default is **false**.

Remarks

In order to use the shape embossment effect, you must also set the EffectsProvider property.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
EnableShadowEffects Property

Diagram.EnableLanes Property

Gets or sets whether lanes are enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableLanes { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableLanes As Boolean
```

Property Value

true if lanes are enabled; otherwise, **false**. The default is **false**.

Remarks

You can customize the lane grid through the LaneGrid property.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.EnableShadowEffects Property

Gets or sets a value indicating whether the shadow dispersion effect for nodes is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableShadowEffects { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableShadowEffects As Boolean
```

Property Value

true to enable shadow dispersion; otherwise, **false**. The default is **false**.

Remarks

In order to use the shadow dispersion effect, you must also set the EffectsProvider property.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
EnableEmbossEffects Property

Diagram.EnableSpatialIndex Property

Gets or sets whether to index item positions for faster hit-testing and viewport clipping queries.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableSpatialIndex { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableSpatialIndex As Boolean
```

Property Value

true to enable spatial indexing, or **false** otherwise. The default value is **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.EnableStyledText Property

Gets or sets the default value for the EnableStyledText property of new nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableStyledText { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableStyledText As Boolean
```

Property Value

true to parse the text of items in order to find formatting tags; otherwise, **false**. The default is **false**.

Remarks

Styled text allows applying various attributes to the characters it contains. Attributes are specified using HTML-like formatting tags embedded in the raw text. Formatting tags are listed and described in the **Text Attributes and Formatting** section of this help file.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ExpandButtonAction Property

Gets or sets a value indicating what should happen when a user clicks the [±] button that is displayed near expandable nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ExpandButtonAction ExpandButtonAction { get; set; }
```

Visual Basic Copy Code

```
Public Property ExpandButtonAction As ExpandButtonAction
```

Property Value

A member of the ExpandButtonAction enumeration. The default is **ExpandTreeBranch**.

Remarks

The default behavior of a [±] button displayed nearby an **Expandable** node is to hide or show the children of that node. In that case, the children of a node are all nodes that can be reached from it by following recursively the **OutgoingLinks** or **IncomingLinks**, as defined by the ExpandOnIncoming property.

To implement custom expand/collapse functionality, set **ExpandButtonAction** to RaiseEvents and handle the ExpandButtonClicked event. That lets you use other criteria for what nodes should be shown or hidden, for example, containment or hierarchical grouping.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ExpandButtonPosition Property

TreeCollapsed Event
TreeExpanded Event

Diagram.ExpandButtonPosition Property

Gets or sets where expand/collapse buttons are located relatively to their nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ExpandButtonPosition ExpandButtonPosition { get; set; }
```

Visual Basic Copy Code

```
Public Property ExpandButtonPosition As ExpandButtonPosition
```

Property Value

A member of the ExpandButtonPosition enumeration. The default is **OuterRight**.

Remarks

If diagram links tend to flow vertically, it would be useful to have the expand/collapse buttons stay at the left- or right-hand side of nodes. If diagram links tend to flow horizontally, choose position for expand buttons such that they stay above or below nodes. That will prevent button icons from being overlapped by links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
NodesExpandable Property
ExpandButtonAction Property

Diagram.ExpandOnIncoming Property

Gets or sets a value, indicating in which direction object hierarchies are expanded.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ExpandOnIncoming { get; set; }
```

Visual Basic Copy Code

```
Public Property ExpandOnIncoming As Boolean
```

Property Value

true if trees are expanded in the direction of the links, which come into nodes; otherwise, **false**. The default is **false**.

Remarks

Use this property when presenting tree structures with enabled collapse/expand. If the property is set to **false**, a clicked expand button of a node expands in the direction of the node's outgoing links. That is, the nodes pointed by links going out of a node are considered children of that node.

See Also

Diagram.ExteriorBrush Property

Gets or sets the brush used for painting the area outside Bounds.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush ExteriorBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property ExteriorBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class. Default value is **null** (**Nothing** in Visual Basic).

Remarks

If **ExteriorBrush** is set to **null**, the BackBrush is used to paint the areas both inside and outside Bounds. In order to scroll the exterior filling and the background image consistently, **ExteriorBrush** can be used only with document-relative BackgroundImageAlign.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Brush and Pen Classes

Diagram.Factory Property

Gets a Factory instance that lets you add programmatically new items to the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Factory Factory { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Factory As Factory
```

Property Value

An instance of the Factory class.

Remarks

You can add a new node or link to the diagram by creating an instance of the item class by using the **new** operator and adding it to the Nodes or Links collection. This works the same way for adding both items from your custom class or ones from the MvcDiagram predefined types. The Factory class methods are just shortcuts that let you do that with a single method call, and they work only for the predefined item classes.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.FocusedComponent Property

Gets or sets the component within this diagram which currently has the input focus.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ComponentBase FocusedComponent { get; set; }
```

Visual Basic Copy Code

```
Public Property FocusedComponent As ComponentBase
```

Property Value

An instance of a ComponentBase -derived class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Components
ComponentBase.GotFocus Event

Diagram.FoldedContainerSize Property

Gets or sets the size of folded containers.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size FoldedContainerSize { get; set; }
```

Visual Basic Copy Code

```
Public Property FoldedContainerSize As Size
```

Property Value

A **Size** instance specifying the size of folded containers. The default value is **Empty**.

Remarks

If this property is not set, the width of folded containers is set to their MinimumSize and the height to their CaptionHeight.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ContainerNode.Folded Property

Diagram.Font Property

Gets or sets the default font for new diagram items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Font Font { get; set; }
```

Visual Basic Copy Code

```
Public Property Font As Font
```

Property Value

An instance of the **Font** class. The default is Microsoft Sans Serif, 9pt.

Remarks

The Font of new nodes and links is initialized with the value of this property.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramItem.Font Property
DiagramView.TextRenderingHint Property

Diagram.FreeFormAttractDistance Property

Gets or sets the radius around dragged free-form adjustment handle in which other points of FreeFormNode are modified too.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double FreeFormAttractDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property FreeFormAttractDistance As Double
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.FreeFormTargets Property

An array of Shape objects used to replace FreeFormNode instances with ShapeNode ones when Behavior is set to LinkFreeShapes or DrawFreeShapes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape[] FreeFormTargets { get; set; }
```

Visual Basic Copy Code

```
Public Property FreeFormTargets As Shape()
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.GridColor Property

Gets or sets the color of the grid points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Color GridColor { get; set; }
```

Visual Basic Copy Code

```
Public Property GridColor As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** instance. The default is Color.FromArgb(140, 140, 150).

Remarks

Use the property to specify the color of grid points and lines.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShowGrid Property
GridSizeX Property
GridSizeY Property
GridStyle Property

Diagram.GridOffsetX Property

Gets or sets the horizontal offset of the first point of the alignment grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double GridOffsetX { get; set; }
```

Visual Basic

```
Public Property GridOffsetX As Double
```

Property Value

A **double** value that specifies the horizontal grid offset. The default is **NaN**.

Remarks

The property specifies the offset of the first point of the alignment grid. The default value **NaN** makes the grid start at offset GridSizeX / 2.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShowGrid Property
GridOffsetY Property
MeasureUnit Property

Diagram.GridOffsetY Property

Gets or sets the vertical offset of the first point of the alignment grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double GridOffsetY { get; set; }
```

Visual Basic

```
Public Property GridOffsetY As Double
```

Property Value

A **double** value that specifies the vertical grid offset. The default is **NaN**.

Remarks

The property specifies the offset of the first point of the alignment grid. The default value **NaN** makes the grid start at offset GridSizeY / 2.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShowGrid Property
GridOffsetX Property
MeasureUnit Property

Diagram.GridPatternHatch Property

Gets or sets a hatch style used to draw dense alignment grid at low zoom levels.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public HatchStyle GridPatternHatch { get; set; }
```

Visual Basic

```
Public Property GridPatternHatch As HatchStyle
```

Property Value

A member of the HatchStyle enumeration.

Remarks

GridPatternHatch is used to draw the alignment grid when its pixels get closer than GridPatternThreshold value due to small **ZoomFactor**. This improves diagram's rendering speed at low zoom levels without having to hide the grid altogether.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShowGrid Property

Diagram.GridPatternThreshold Property

Gets or sets the distance between grid points (in pixels) that triggers drawing using GridPatternHatch.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double GridPatternThreshold { get; set; }
```

Visual Basic

```
Public Property GridPatternThreshold As Double
```

Property Value

A number specifying distance between zoomed grid points. The default value is **2.5**.

Remarks

GridPatternHatch is used to draw the alignment grid when its pixels get closer than **GridPatternThreshold** value due to small **ZoomFactor**. This improves diagram's rendering speed at low zoom levels without having to hide the grid altogether.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShowGrid Property

Diagram.GridSizeX Property

Gets or sets the horizontal distance between adjacent grid points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double GridSizeX { get; set; }
```

Visual Basic

```
Public Property GridSizeX As Double
```

Property Value

A **double** value specifying the horizontal distance between grid points. The default is 4.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShowGrid Property
AlignToGrid Property
AlignPointToGrid Method
MeasureUnit Property

Diagram.GridSizeY Property

Gets or sets the vertical distance between adjacent grid points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double GridSizeY { get; set; }
```

Visual Basic Copy Code

```
Public Property GridSizeY As Double
```

Property Value

A **double** value specifying the vertical distance between grid points. The default is 4.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShowGrid Property
AlignToGrid Property
AlignPointToGrid Method
MeasureUnit Property

Diagram.GridStyle Property

Gets or sets the visual style of the alignment grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public GridStyle GridStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property GridStyle As GridStyle
```

Property Value

A member of the GridStyle enumeration. The default is **Points**.

Remarks

Specifies how the alignment grid is drawn. If the property value is set to Points, the grid is painted as a matrix of points. If set to **Lines**, the grid is represented as a series of crossing horizontal and vertical lines. Grid points or lines are painted with the color set in GridColor; the distance between them is specified by the GridSizeX and GridSizeY properties.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShowGrid Property
AlignToGrid Property

Diagram.Groups Property

Gets a collection of all groups in the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public GroupCollection Groups { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Groups As GroupCollection
```

Property Value

An instance of the GroupCollection class.

Remarks

Groups provides access to all groups defined in the current diagram.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Items Property
Factory.CreateGroup Method

Diagram.HitTestPriority Property

Gets or sets a value indicating the order in which items are hit-tested.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HitTestPriority HitTestPriority { get; set; }
```

Visual Basic Copy Code

```
Public Property HitTestPriority As HitTestPriority
```

Property Value

A member of the HitTestPriority enumeration. The default value is **NodesBeforeLinks**.

Remarks

For easier selection, links have a hit-detection area several pixels wider than their visible segment lines. However, if several links with higher z-index cross a node, it would be difficult to select it. The **HitTestPriority** property allows altering hit-test results in such situations. If set to NodesBeforeLinks, nodes are hit-tested before all links. **ZOrder** makes all items tested in decreasing z-order, independently of their type.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LinkHitDistance Property

Diagram.Interaction Property

Gets an InteractionState instance representing the current action being carried out by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public InteractionState Interaction { get; set; }
```

Visual Basic Copy Code

```
Public Property Interaction As InteractionState
```

Property Value

An instance of the InteractionState class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.IsProEdition Property

Gets a value indicating whether the currently loaded MvcDiagram is a professional version of the control.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsProEdition { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property IsProEdition As Boolean
```

Property Value

true if the currently loaded MvcDiagram control is a professional version; otherwise, **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.Items Property

Gets a collection of all items in this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItemCollection Items { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Items As DiagramItemCollection
```

Property Value

An instance of the DiagramItemCollection class.

Remarks

This collection gives you access to all items contained in this document. You can also use it to add new items to the diagram.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Nodes Property
Links Property

Diagram.LaneGrid Property

Gets the Grid instance that represents the lane grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Grid LaneGrid { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property LaneGrid As Grid
```

Property Value

An instance of the Grid class.

Remarks

The Grid object gives you full control over the lane grid. For example, it lets you set the number of rows and columns in the grid, add headers, customize the appearance, subscribe to events and so on.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.Layers Property

Gets a list of the layers in the current diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IList<Layer> Layers { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Layers As IList(Of Layer)
```

Property Value

A list with all diagram layers.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.LinkBaseShape Property

Gets or sets the shape displayed at the base of new links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape LinkBaseShape { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkBaseShape As Shape
```

Property Value

A Shape instance that represents an arrowhead shape. The default is None.

Remarks

This property specifies the default shape displayed as arrowhead at the links' origin points. The ArrowHeads class exposes several predefined shapes as static properties.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.BaseShape Property
LinkBaseShapeSize Property
LinkHeadShape Property
LinkIntermediateShape Property

Diagram.LinkBaseShapeSize Property

Gets or sets the default size of the base of new links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double LinkBaseShapeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkBaseShapeSize As Double
```

Property Value

A **double** value that specifies the size. The default is 5.

Remarks

The property specifies the size of the graphical shapes that are displayed at the start of new diagram links, for example, at the beginning of the links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.BaseShapeSize Property
LinkBaseShape Property

Diagram.LinkBranchIndicator Property

Gets or sets a value indicating what shapes to display as markers at points where overlapping Cascading links diverge.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public BranchIndicator LinkBranchIndicator { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkBranchIndicator As BranchIndicator
```

Property Value

A member of the BranchIndicator enumeration. The default value is **None**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
BranchIndicatorSize Property
BranchIndicatorColor Property
LinkCrossings Property
LinkMergeIndicator Property
DrawBranchIndicator Event

Diagram.LinkBrush Property

Gets or sets the default value for the Brush property of links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush LinkBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

Remarks

The property provides a default value for the brush, with which an arrowhead's interior is filled. The Brush property of new links is initialized with a clone of **LinkBrush**, but after link creation the properties are independent - changing one of them does not change the other.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Brush and Pen Classes
LinkPen Property

Diagram.LinkCascadeOrientation Property

Gets or sets the initial orientation of the first segment of new cascading links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Orientation LinkCascadeOrientation { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkCascadeOrientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default value is **Auto**.

Remarks

Use this property to specify the default orientation of the first segment of DiagramLink objects whose Shape is set to LinkShape.Cascading. The default value, Auto, lets the component decide whether to draw the first segment horizontally or vertically. See **Orientation** for more details.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.CascadeOrientation Property

Diagram.LinkCrossings Property

Gets or sets how to render the intersection points where links cross their paths.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkCrossings LinkCrossings { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkCrossings As LinkCrossings
```

Property Value

A member of the LinkCrossings enumeration. The default is **Straight**.

Remarks

Intersection points where links cross each other's paths can be rendered as arcs or breaks, depending on the value of this property. The z-order defines which links jump over the others or which links are cut by the others. If **LinkCrossings** is set to Arcs, links with higher ZIndex jump over those with lower **ZIndex**. If set to **Cut**, segments of links at lower z-position are cut by links at higher z-position. Radius of arcs and breaks can be set by the CrossingRadius property.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.DrawCrossings Property
RouteLinks Property
RoundedLinks Property
LinkBranchIndicator Property
LinkMergeIndicator Property
ValidateLinkCrossing Event

Diagram.LinkCustomDraw Property

Gets or sets the default type of custom painting done on links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CustomDraw LinkCustomDraw { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkCustomDraw As CustomDraw
```

Property Value

A member of the CustomDraw enumeration. The default is **None**.

Remarks

This property specifies whether and how custom painting of links is performed. Its value is assigned to the CustomDraw property of new links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DrawLink Event

Diagram.LinkEndsMovable Property

Gets or sets a value indicating whether link ends can be moved by users after the link is created.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool LinkEndsMovable { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkEndsMovable As Boolean
```

Property Value

true if moving link ends is allowed; **false**, if not.

Remarks

When the property is set to **true**, users cannot move the ends of a link after the link is created. This prevents them from connecting the links to other origin and destination nodes.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LinkModifying Event

Diagram.LinkHandlesStyle Property

Gets or sets the default value for the HandlesStyle property of new links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HandlesStyle LinkHandlesStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkHandlesStyle As HandlesStyle
```

Property Value

A member of the HandlesStyle enumeration. The default is **SquareHandles**.

Remarks

Default value for the HandlesStyle property of new links, created either programmatically or by a user. That property specifies how links' selection status is indicated visually.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.LinkHeadShape Property

Gets or sets the shape displayed as head of new links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape LinkHeadShape { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkHeadShape As Shape
```

Property Value

A Shape instance that represents an arrowhead shape. The default is Arrow.

Remarks

This property specifies the default shape displayed as arrowhead at the links' end points. The ArrowHeads class exposes several predefined shapes as static properties.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.HeadShape Property
LinkHeadShapeSize Property
LinkBaseShape Property
LinkIntermediateShape Property

Diagram.LinkHeadShapeSize Property

Gets or sets the default size of arrowheads.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double LinkHeadShapeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkHeadShapeSize As Double
```

Property Value

A **double** value that measures the size. The default is 5.

Remarks

The property specifies how big the shapes of new arrowheads are.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.HeadShapeSize Property
LinkHeadShape Property

Diagram.LinkHitDistance Property

Gets or sets a value specifying how far from a link a click is still considered a hit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double LinkHitDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkHitDistance As Double
```

Property Value

A **double** value specifying the distance in MeasureUnit units. The default value is 0.

Remarks

If this property is set to 0, the control considers it a hit when a link is no further than five millimeters from the mouse position. This **LinkHitDistance** value is also used when hit-testing programmatically, e.g. by calling the GetItemAt method.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
HitTestPriority Property
AutoSnapDistance Property

Diagram.LinkIntermediateShape Property

Gets or sets the default shape to be displayed in the middle of each segment of links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape LinkIntermediateShape { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkIntermediateShape As Shape
```

Property Value

A Shape instance that represents an arrowhead shape. The default is None.

Remarks

The property specifies default shape to be displayed at the middle of each segment of new links. It facilitates users by following the path of a link across the diagram, especially for longer links in large documents. The ArrowHeads class exposes several predefined shapes as static properties.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.IntermediateShape Property
LinkIntermediateShapeSize Property

LinkHeadShape Property
LinkBaseShape Property

Diagram.LinkIntermediateShapeSize Property

Gets or sets the default size of the shapes at middles of link segments.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double LinkIntermediateShapeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkIntermediateShapeSize As Double
```

Property Value

A **double** value that measures the size. The default is 5.

Remarks

Specifies default value for the IntermediateShapeSize property of new links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.IntermediateShapeSize Property
LinkIntermediateShape Property

Diagram.LinkMergeIndicator Property

Gets or sets a value indicating what shapes to display as markers at points where overlapping Cascading links merge.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public BranchIndicator LinkMergeIndicator { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkMergeIndicator As BranchIndicator
```

Property Value

A member of the BranchIndicator enumeration. The default value is **None**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
MergeIndicatorSize Property
MergeIndicatorColor Property
LinkBranchIndicator Property
DrawBranchIndicator Event

Diagram.LinkPen Property

Gets or sets the default value for the Pen and HeadPen properties of links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen LinkPen { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkPen As Pen
```

Property Value

An instance of the MindFusion.Drawing.Pen class.

Remarks

The property provides a default value for the pens, with which link's segments and arrowhead lines are painted. The Pen and HeadPen properties of new links are initialized with clones of **LinkPen**, but after link creation the properties are independent - changing one of them does not change the others.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Brush and Pen Classes
LinkBrush Property

Diagram.LinkRouter Property

Gets or sets the component used to find paths for auto-routed links when diagram nodes are added or their positions change.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ILinkRouter LinkRouter { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkRouter As ILinkRouter
```

Property Value

An instance of a class implementing the ILinkRouter interface.

Remarks

The default value is a CompositeRouter instance that chains PatternRouter and GridRouter.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AutoRoute Property
RouteLinks Property
RouteAllLinks Method
CompositeRouter
GridRouter

PatternRouter
QuickRouter

Diagram.Links Property

Gets a collection of all DiagramLink objects in the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection Links { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Links As DiagramLinkCollection
```

Property Value

An instance of the DiagramLinkCollection class.

Remarks

Links provides type-safe access to all links in the current diagram. Any diagram element can be accessed also via the Items collection.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Items Property
Nodes Property
Groups Property
Factory.CreateDiagramLink Method

Diagram.LinkSegments Property

Gets or sets the default number of link segments.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int LinkSegments { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkSegments As Integer
```

Property Value

An integer value specifying how many segments links consists of. The default is 1.

Remarks

This value is assigned to the SegmentCount property of new links, which together with Shape defines the number of ControlPoints. You can let users insert new link segments interactively by setting AllowSplitLinks to **true**.

See Also

Diagram.LinkShape Property

Gets or sets the default value of the Shape property of new links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkShape LinkShape { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkShape As LinkShape
```

Property Value

A member of the LinkShape enumeration. The default value is **Polyline**.

Remarks

Use this property to specify what kind of segments new links are composed of. Curved links can be created using Bezier or **Spline** styles. Straight-segment links can be created using **Polyline** and **Cascading** styles, the latter one restricting segments to horizontal and vertical direction, orienting each segment orthogonally to its neighbor.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.Shape Property
RoundedLinks Property
LinkBranchIndicator Property

Diagram.LinksRetainForm Property

Gets or sets whether links should maintain their forms if the nodes linked with those links are moved around.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool LinksRetainForm { get; set; }
```

Visual Basic Copy Code

```
Public Property LinksRetainForm As Boolean
```

Property Value

true to retain links' forms; otherwise, **false**. The default is **false**.

Remarks

LinksRetainForm value is used to initialize the RetainForm property of new links. It has effect on links that have more than two control points. If the nodes that a link connects are moved around, usually only the segments at the ends of that link are updated, so that they stay connected to their respective nodes. In that case all intermediate control points stay where they are. However, if

RetainForm is enabled, all control points will move simultaneously in a way they preserve their relative position to each other, and thus preserving the overall form of the link.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
RouteLinks Property
DynamicLinks Property

Diagram.LinkText Property

Gets or sets the default text of links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string LinkText { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkText As String
```

Property Value

A string specifying the initial text of links. The default is an empty string.

Remarks

The value of this property is assigned to the Text property of new links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.TextColor Property
TextFormat Property

Diagram.LinkTextStyle Property

Gets or sets default placement and orientation for links' text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkTextStyle LinkTextStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkTextStyle As LinkTextStyle
```

Property Value

A member of the LinkTextStyle enumeration. The default value is **Center**.

Remarks

The value of this property is used to initialize the TextStyle property of new links.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.MeasureTextRenderingHint Property

Gets or sets the TextRenderingHint to use when measuring texts. This should be effectively set to the value of DiagramView.TextRenderingHint.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static TextRenderingHint MeasureTextRenderingHint { get; set; }
```

Visual Basic Copy Code

```
Public Shared Property MeasureTextRenderingHint As TextRenderingHint
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.MeasureUnit Property

Gets or sets the unit of measure used for logical coordinates in this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MeasureUnit MeasureUnit { get; set; }
```

Visual Basic Copy Code

```
Public Property MeasureUnit As MeasureUnit
```

Property Value

An instance of the MeasureUnit class. The default value is Millimeter.

Remarks

Many properties in the MvcDiagram API are used to specify the dimensions of diagram items or the size of various visual elements, for example, selection handles, arrowhead shapes, frame widths, and so on. **MeasureUnit** specifies the measure of length in which these size values are defined.

Changing the measure unit does not automatically rescale the size-related property values. Your application must set appropriate values to: Bounds, GridSizeX, GridSizeY, GridOffsetX, GridOffsetY, AdjustmentHandlesSize, LinkBaseShapeSize, LinkHeadShapeSize, LinkIntermediateShapeSize, TableCaptionHeight, TableColumnWidth, TableRowHeight, CrossingRadius, ShadowOffsetX, ShadowOffsetY and to the size-related members of RoutingOptions.

Changing the measure unit resets **RoutingOptions** to default values for the new unit, because performance might suffer greatly if the routing algorithm uses too big or too small GridSize.

The arguments passed to some methods depend on the measure unit too, for example, CreateShapeNode, CreateTableNode, GetNodeAt, GetLinkAt, **ZoomToRect**.

The following MeasureUnit members provide ready-to-use measure units:

Member name	Description
Centimeter	Specifies centimeters as the unit of measure.
Document	Specifies the document unit (1/300 inch) as the unit of measure.
Inch	Specifies the inch as the unit of measure.
Millimeter	Specifies the millimeter as the unit of measure.
Pixel	Specifies a device pixel as the unit of measure. Note that print and on-screen output will differ greatly in size when using pixels because printer pixels are much smaller than screen ones.
Point	Specifies a printer's point (1/72 inch) as the unit of measure.
WpfPoint	Specifies device-independent pixels (1/96 inch) as the unit of measure.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ChangeUnit Method

Diagram.MergeIndicatorColor Property

Gets or sets the color of link merge indicators.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Color MergeIndicatorColor { get; set; }
```

Visual Basic Copy Code

```
Public Property MergeIndicatorColor As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** instance specifying the color of branch indicators. The default value is **LightGreen**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LinkMergeIndicator Property
MergeIndicatorSize Property

Diagram.MergeIndicatorSize Property

Gets or sets the size of link merge indicators.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double MergeIndicatorSize { get; set; }
```

Visual Basic Copy Code

```
Public Property MergeIndicatorSize As Double
```

Property Value

A **double** value specifying the size of branch indicators. The default value is 4.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LinkMergeIndicator Property
MergeIndicatorColor Property

Diagram.MergeThreshold Property

Gets or sets the maximum distance between adjacent control points of a link at which the respective segments can be merged.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double MergeThreshold { get; set; }
```

Visual Basic Copy Code

```
Public Property MergeThreshold As Double
```

Property Value

A **double** value Specifying the maximum distance in the current MeasureUnit. The default is 0.

Remarks

A user may add or remove links' segments interactively when the AllowSplitLinks property is enabled. They can drag a link's control point over the next one in order to merge two adjacent segments. Usually the two points must be placed exactly at the same coordinates in order to merge the respective segments of the link. Assign a larger value to this property if you wish to make it easier for users to merge segments by just moving link's control point close to each other.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.MinimumNodeSize Property

Gets or sets the minimum node size that the user is allowed to set.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size MinimumNodeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property MinimumNodeSize As Size
```

Property Value

A **Size** instance specifying the minimum size allowed for nodes.

Remarks

If the user releases the mouse button while creating or modifying a node and the node size is smaller than this value, the operation is cancelled.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ModificationEffect Property

Gets or sets the visual effect to apply during interactive modification of items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ModificationEffect ModificationEffect { get; set; }
```

Visual Basic Copy Code

```
Public Property ModificationEffect As ModificationEffect
```

Property Value

A member of the ModificationEffect enumeration. The default value is **None**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.NodeEffects Property

Gets a collection with all effects applied to the nodes of this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IList<NodeEffect> NodeEffects { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NodeEffects As IList(Of NodeEffect)
```

Property Value

A list with all currently active node effects.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.Nodes Property

Gets a collection of all DiagramNode objects in the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeCollection Nodes { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Nodes As DiagramNodeCollection
```

Property Value

An instance of the DiagramNodeCollection class.

Remarks

Nodes provides type-safe access to all nodes in the current diagram. Any diagram element can be accessed also by using the Items collection.

Example

The following code adds a new ShapeNode to the diagram:

C# Copy Code

```
var node = new ShapeNode(diagram);
node.Bounds = new Rect(10, 10, 30, 30);
diagram.Nodes.Add(node);
```

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Items Property
Links Property
Groups Property
Factory.CreateShapeNode Method

Diagram.NodesExpandable Property

Gets or sets a value indicating whether newly created nodes can be expanded.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool NodesExpandable { get; set; }
```

Visual Basic Copy Code

```
Public Property NodesExpandable As Boolean
```

Property Value

true if new nodes can be expanded; otherwise, **false**. The default is **false**.

Remarks

If using MvcDiagram to present tree/hierarchical diagrams, you might set **NodesExpandable** to **true**. If enabled, the property makes each newly created node appear with a [±] sign next to it, which expands or collapses the node's descendants.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ExpandButtonPosition Property
RecursiveExpand Property
ExpandOnIncoming Property
TreeCollapsed Event
TreeExpanded Event
DiagramNode.Expandable Property

Diagram.NowLoading Property

Gets a value indicating whether the diagram data is now being loaded from a file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool NowLoading { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NowLoading As Boolean
```

Property Value

true if the diagram is currently in a loading state; otherwise, **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
LoadFromFile Method

Diagram.PolygonalTextLayout Property

Gets or sets a value indicating whether to enable the formatting of text inside non-rectangular shape node outlines.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool PolygonalTextLayout { get; set; }
```

Visual Basic Copy Code

```
Public Property PolygonalTextLayout As Boolean
```

Property Value

true to enable polygonal formatting; **false** to disable it.

Remarks

Specifies a default value for the PolygonalTextLayout property of shape nodes.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DefaultShape Property

Diagram.RecursiveExpand Property

Gets or sets value indicating how a tree is expanded.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RecursiveExpand { get; set; }
```

Visual Basic Copy Code

```
Public Property RecursiveExpand As Boolean
```

Property Value

true if a tree is recursively expanded; otherwise, **false**. The default is **false**.

Remarks

Use the property when representing tree structures, which hide some of their elements. When the property is set to **true**, clicking on a certain node in the tree, expands not only its descendants, but all descendants of the descendants, and so on, till the whole tree is expanded.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ExpandOnIncoming Property
NodesExpandable Property

Diagram.RenderOptions Property

Gets the current rendering options for this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RenderOptions RenderOptions { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property RenderOptions As RenderOptions
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.RestrictItemsToBounds Property

Gets or sets a value indicating whether users are allowed to place items outside the scrollable area of the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RestrictToBounds RestrictItemsToBounds { get; set; }
```

Visual Basic Copy Code

```
Public Property RestrictItemsToBounds As RestrictToBounds
```

Property Value

A member of the RestrictToBounds enumeration. The default is **Intersection**.

Remarks

Use this property to specify whether diagram elements can be drawn or moved outside the document area. It is possible to specify that items have to be entirely inside, partly outside or there is no restriction.

If there is no restriction and AutoResize is disabled, users could place an item to a position where they cannot reach it by scrolling. That might be interpreted as an item deletion operation by handling the NodeModified event and deleting the item if it is outside the document boundaries.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Bounds Property
AutoResize Property

Diagram.RoundedLinks Property

Gets or sets a value indicating whether the joins between segments of Polyline and **Cascading** links should be rounded.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RoundedLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property RoundedLinks As Boolean
```

Property Value

true to use rounded segment joins; otherwise, **false**. The default is **false**.

Remarks

The value of this property is considered when drawing links whose Shape is set to Polyline or **Cascading**. If the property is enabled, the joins between the segments of all such links are drawn as arcs, whose radius equals the value of RoundedLinksRadius.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramLink.Shape Property
LinkCrossings Property

Diagram.RoundedLinksRadius Property

Gets or sets the radius of arc joins of rounded links' segments.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double RoundedLinksRadius { get; set; }
```

Visual Basic Copy Code

```
Public Property RoundedLinksRadius As Double
```

Property Value

A **double** value specifying the radius length. The default is 2.

Remarks

The value of this property is used when RoundedLinks is enabled. It specifies the radius of the arcs drawn as joins between link segments.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
CrossingRadius Property

Diagram.RoundRectFactor Property

Gets or sets the relative size of arcs displayed at the corners of rounded rectangles.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double RoundRectFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property RoundRectFactor As Double
```

Property Value

A multiplier applied to the radius of the arcs used as corners of rounded rectangles. The default is 1.

Remarks

The default length of the radius of rounded-corner arcs is 7. This property lets you increase or decrease that value by multiplying it by the specified factor.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
TableShape Property

Diagram.RouteLinks Property

Gets or sets a value indicating whether links should avoid nodes by going the shortest path from their origin to their destination without crossing any other nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RouteLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property RouteLinks As Boolean
```

Property Value

true to route links; otherwise, **false**. The default is **false**.

Remarks

The value of **RouteLinks** is used to initialize the AutoRoute property of new links. If a node is moved or resized, all links that intersect with its new position and have **AutoRoute** enabled are rerouted. Keep in mind that number of link segments will probably change when routing is enabled.

RoutingOptions allows tuning the routing algorithm. The most important options are the cost values exposed as LengthCost, TurnCost and CrossingCost properties. They greatly influence the shape of the generated route.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
RouteAllLinks Method
DynamicLinks Property
LinkCrossings Property
LinkRouter Property

Diagram.RoutingOptions Property

Gets an object, which allows customizing various parameters of the link routing algorithm.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RoutingOptions RoutingOptions { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property RoutingOptions As RoutingOptions
```

Property Value

An instance of the RoutingOptions class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
RouteLinks Property
RouteAllLinks Method
LinkRouter Property

Diagram.RowHighlightBrush Property

Gets or sets the Brush used to fill highlighted table rows.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush RowHighlightBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property RowHighlightBrush As Brush
```

Property Value

An instance of a Brush subclass.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ScriptHelper Property

Gets a ScriptHelper object which can be used in html scripts to instantiate various objects and to call static methods.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ScriptHelper ScriptHelper { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ScriptHelper As ScriptHelper
```

Property Value

An instance of the ScriptHelper class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SelectAfterCreate Property

Gets or sets a value indicating whether newly created objects are automatically selected.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool SelectAfterCreate { get; set; }
```

Visual Basic Copy Code

```
Public Property SelectAfterCreate As Boolean
```

Property Value

true if items are selected after being created; otherwise, **false**. The default is **true**.

Remarks

Use the property to specify if the items that are drawn are selecting, which allows immediate modification actions to be performed over them.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Selection Property

Diagram.SelectedItemHandlesStyle Property

Gets the visual style of adjustment handles of selected items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HandlesVisualStyle SelectedItemHandlesStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property SelectedItemHandlesStyle As HandlesVisualStyle
```

Property Value

An instance of the HandlesVisualStyle class.

Remarks

This style object provides Pen and Brush attributes that let you customize the appearance of adjustment handles drawn around selected items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Selection Property
ActiveItemHandlesStyle Property

Diagram.Selection Property

Gets a Selection instance that represents the selection of items in this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Selection Selection { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Selection As Selection
```

Property Value

An instance of the Selection class representing the current selection.

Remarks

Use this property to get the items currently selected by a user. Items can be enumerated by the Items member of the Selection class. An item from the selection is considered the active one and can be accessed by using ActiveItem. To prevent users from selecting multiple items, set AllowMultipleSelection to **false**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramView.ModificationStart Property
AdjustmentHandlesSize Property
HitTestPriority Property

Diagram.SelectionOnTop Property

Gets or sets a value indicating whether selected items should be painted on top of other items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool SelectionOnTop { get; set; }
```

Visual Basic Copy Code

```
Public Property SelectionOnTop As Boolean
```

Property Value

true if selected items should be painted on top; otherwise, **false**. The default is **true**.

Remarks

If **SelectionOnTop** is enabled, selected items are painted over non-selected ones, ignoring the z-order position specified in ZIndex. If disabled, the items are painted as set in the z-order.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Selection Class

ActiveItem Property
SortGroupsByZ Property

Diagram.SerializationBinder Property

Gets or sets a SerializationBinder object used to control class loading.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SerializationBinder SerializationBinder { get; set; }
```

Visual Basic Copy Code

```
Public Property SerializationBinder As SerializationBinder
```

Property Value

An instance of **System.Runtime.Serialization.SerializationBinder** -derived class. The default value is **null** (**Nothing** in VB.NET).

Remarks

You can use a custom serialization binder when serializable classes used in your application have been renamed or moved to a different namespace. Objects from such classes cannot be deserialized, unless the **BindToType** method of **SerializationBinder** maps the old class names to their corresponding types. Return **null** from your **BindToType** implementation for types you don't recognize, to let the diagram control process its own obsolete types.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ShadowBrush Property

Gets or sets the default brush used to paint the shadows cast by diagram items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush ShadowBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property ShadowBrush As Brush
```

Property Value

An instance of a Brush -derived class. The default is SolidBrush(Color.FromArgb(150, Color.DarkGray)).

Remarks

Use the property to specify an initial value for the ShadowBrush property of new diagram elements.

See Also

Diagram Members
Diagram Class

Diagram.ShadowOffsetX Property

Gets or sets the horizontal offset of item shadows.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ShadowOffsetX { get; set; }
```

Visual Basic Copy Code

```
Public Property ShadowOffsetX As Double
```

Property Value

A **double** value that specifies the X shadow offset.

Remarks

Use the property to specify how distant the shadow of the item should be drawing as to the x-coordinate axis. You can set both X and Y offset values with the SetShadowOffset method.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
SetShadowOffset Method
DiagramItem.ShadowOffsetY Property

Diagram.ShadowOffsetY Property

Gets or sets the vertical offset of item shadows.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ShadowOffsetY { get; set; }
```

Visual Basic Copy Code

```
Public Property ShadowOffsetY As Double
```

Property Value

A **double** value that specifies the Y shadow offset.

Remarks

Use the property to specify how distant the shadow of the item should be drawing as to the y-coordinate axis. You can set both X and Y offset values with the SetShadowOffset method.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

SetShadowOffset Method
DiagramItem.ShadowOffsetX Property

Diagram.ShadowsStyle Property

Gets or sets the style for rendering shadows.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShadowsStyle ShadowsStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property ShadowsStyle As ShadowsStyle
```

Property Value

A member of the ShadowsStyle enumeration.

Remarks

This property specifies how the shadows of diagram items are drawn. If its value is None no shadows are painted. **OneLevel** makes all shadows paint on the lowest z-level, not covering any items. Thus the diagram items will seem to appear on the same z-level. **ZOrder** makes the shadow paint immediately under their items, possibly covering other items. This style gives more 3D-like outlook to the diagram.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramItem.ShadowOffsetX Property
DiagramItem.ShadowOffsetY Property
EnableShadowEffects Property

Diagram.ShapeBrush Property

Gets or sets the default value for the Brush property of shape nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush ShapeBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property ShapeBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

Remarks

The property provides a default value for the brush, with which a node' interior is filled. The Brush property of new shape nodes is initialized with a clone of **ShapeBrush**, but after the node creation the properties are independent - changing one of them does not change the other.

In **ImageMap** mode of NetDiagram, this property also specifies the color that fills the DIV element representing the new node being drawn.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Brush and Pen Classes
ShapePen Property

Diagram.ShapeCustomDraw Property

Gets or sets the default type of custom painting done for shape nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CustomDraw ShapeCustomDraw { get; set; }
```

Visual Basic Copy Code

```
Public Property ShapeCustomDraw As CustomDraw
```

Property Value

A member of the CustomDraw enumeration. The default is **None**.

Remarks

This property specifies whether and how custom painting of shape nodes is performed. Its value is assigned to the CustomDraw property of new shape nodes.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DrawNode Event

Diagram.ShapeHandlesStyle Property

Gets or sets the default value for the **HandlesStyle** property of new shape nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HandlesStyle ShapeHandlesStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property ShapeHandlesStyle As HandlesStyle
```

Property Value

A member of the HandlesStyle enumeration.

Remarks

This property provides default value for the **HandlesStyle** property of new shape nodes, created either programmatically or by the user. That property specifies how selected nodes are indicated visually, and how they can be moved, resized or connected to other objects.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ShapeNodeStyle Property

Gets the theme-level style for shape nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNodeStyle ShapeNodeStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ShapeNodeStyle As ShapeNodeStyle
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ShapeOrientation Property

Gets or sets the default value for the ShapeOrientation property of shape nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ShapeOrientation { get; set; }
```

Visual Basic Copy Code

```
Public Property ShapeOrientation As Double
```

Property Value

A **double** value specifying an angle in degrees. Allowed values are 0, 90, 180 and 270.

Remarks

Use this property in conjunction with DefaultShape to define what shape will be assigned to new nodes that end-users draw.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShapeNode.ShapeOrientation Property
DefaultShape Property

Diagram.ShapePen Property

Gets or sets the default value for the Pen property of shape nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen ShapePen { get; set; }
```

Visual Basic Copy Code

```
Public Property ShapePen As Pen
```

Property Value

An instance of the MindFusion.Drawing.Pen class.

Remarks

The property provides a default value for the pen, with which a shape node's frame is painted. The Pen property of new shape nodes is initialized with a clone of **ShapePen**, but after the node creation the properties are independent - changing one of them does not change the other.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Brush and Pen Classes
ShapeBrush Property

Diagram.ShapeText Property

Gets or sets the default text of shape nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ShapeText { get; set; }
```

Visual Basic Copy Code

```
Public Property ShapeText As String
```

Property Value

A string specifying the initial text of shape nodes. The default is an empty string.

Remarks

The value of this property is assigned to the Text property of new nodes.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
TextColor Property
TextFormat Property

Diagram.ShowAnchors Property

Gets or sets a value indicating whether and when node anchor points should be displayed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShowAnchors ShowAnchors { get; set; }
```

Visual Basic Copy Code

```
Public Property ShowAnchors As ShowAnchors
```

Property Value

A member of the ShowAnchors enumeration. The default value is **Auto**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
SnapToAnchor Property
Anchor Points and Patterns

Diagram.ShowDisabledHandles Property

Gets or sets a value indicating whether manipulation handles are painted when they are disabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ShowDisabledHandles { get; set; }
```

Visual Basic Copy Code

```
Public Property ShowDisabledHandles As Boolean
```

Property Value

true to display disabled manipulation handles; otherwise, **false**. The default is **true**.

Remarks

Individual handles can be disabled or enabled by clearing or setting their correspondent bits in node's **EnabledHandles**. If disabled handles are displayed, they are painted using the HandleBrush specified through DisabledHandlesStyle.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ShowGrid Property

Gets or sets a value indicating whether the alignment grid is visible.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ShowGrid { get; set; }
```

Visual Basic Copy Code

```
Public Property ShowGrid As Boolean
```

Property Value

true if the grid is visible; otherwise, **false**. The default is **false**.

Remarks

If it is visible, the grid is painted as either a matrix of points or crossing horizontal and vertical lines. That depends on the value of GridStyle.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
AlignToGrid Property
GridColor Property
GridSizeX Property
GridSizeY Property
GridPatternHatch Property

Diagram.ShowHandlesOnDrag Property

Gets or sets a value indicating whether to display selection handles of objects under the mouse while another object is being modified.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ShowHandlesOnDrag { get; set; }
```

Visual Basic Copy Code

```
Public Property ShowHandlesOnDrag As Boolean
```

Property Value

true to display the selection handles of an item while it is being modified; **false** to hide them.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ModificationEffect Property

Diagram.SnapToAnchor Property

Gets or sets when link end points should be aligned to the nearest anchor points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SnapToAnchor SnapToAnchor { get; set; }
```

Visual Basic Copy Code

```
Public Property SnapToAnchor As SnapToAnchor
```

Property Value

A member of the SnapToAnchor enumeration. The default is **OnCreate**.

Remarks

The default value OnCreate, specifies that links are snapped to anchor points when they are created, but allows the user to move the link ends to different locations of a node. The **OnCreateOrModify** value makes link ends points connect to node anchors both if links are created or modified.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
ShowAnchors Property

Diagram.SortGroupsByZ Property

Gets or sets a value indicating whether grouped items should be drawn sorted by their z-order while the whole group is being moved.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool SortGroupsByZ { get; set; }
```

Visual Basic Copy Code

```
Public Property SortGroupsByZ As Boolean
```

Property Value

true if grouped items should be painted according to their z-order; otherwise, **false**. The default is **false**.

Remarks

By default, while a group of nodes is being modified with the mouse, the grouped items are drawn in the order defined by the group hierarchy. For example, the group MainItem is drawn bellow all attached items, and if an attached item is the main item in another group, it is drawn below the next groups in the hierarchy. During this process, the ZIndex of items is ignored and the group appearance while it is being modified might differ from how it looks usually. The **SortGroupsByZ** property lets you override the default behavior and display items sorted by their z-order. Take heed if your application creates large hierarchies of groups - enabling that property might slow down the modification process.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
SelectionOnTop Property

Diagram.Style Property

Gets or sets the style associated with the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramStyle Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As DiagramStyle
```

Property Value

An instance of the DiagramStyle class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.SvgNodeStyle Property

Gets the theme-level style for SVG nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNodeStyle SvgNodeStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property SvgNodeStyle As SvgNodeStyle
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.TableBrush Property

Gets or sets the default value for the Brush property of tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush TableBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property TableBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush -derived class.

Remarks

The property provides a default value for the brush, with which a table' interior is filled. The Brush property of new tables is initialized with a clone of **TableBrush**, but after table creation the properties are independent - changing one of them does not change the other.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Brush and Pen Classes
TablePen Property

Diagram.TableCaption Property

Gets or sets the default text for tables' captions.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string TableCaption { get; set; }
```

Visual Basic Copy Code

```
Public Property TableCaption As String
```

Property Value

A string specifying the initial text of table captions. The default is "Table".

Remarks

The Caption property of new tables is initialized with the value of **TableCaption**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
TableCaptionHeight Property

Diagram.TableCaptionHeight Property

Gets or sets the default height of table captions.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double TableCaptionHeight { get; set; }
```

Visual Basic Copy Code

```
Public Property TableCaptionHeight As Double
```

Property Value

A **double** value, which indicates the height of table caption. Default value is 5.

Remarks

The value of this property is used to initialize the CaptionHeight property of new tables.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
TableCaption Property

Diagram.TableColumnCount Property

Gets or sets the default number of table columns.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int TableColumnCount { get; set; }
```

Visual Basic Copy Code

```
Public Property TableColumnCount As Integer
```

Property Value

An integer value, which specifies the number of columns in new tables. The default is 2.

Remarks

This value is used to initialize the ColumnCount property of new tables.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
TableRowCount Property

Diagram.TableColumnWidth Property

Gets or sets the default width of tables' columns.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double TableColumnWidth { get; set; }
```

Visual Basic Copy Code

```
Public Property TableColumnWidth As Double
```

Property Value

A **double** value, which specifies the width of table columns measured. The default is 18.

Remarks

This property is used to initialize the ColumnWidth property of new tables.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
TableRowHeight Property

Diagram.TableConnectionStyle Property

Gets or sets a value indicating whether drawing a link between tables should connect the rows of those tables, or the tables as whole entities.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableConnectionStyle TableConnectionStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property TableConnectionStyle As TableConnectionStyle
```

Property Value

A member of the TableConnectionStyle enumeration. The default is **Rows**.

Remarks

The value of **TableConnectionStyle** is used to initialize the ConnectionStyle property of new tables. There are three possible ways to relate tables to other nodes. This property specifies how users are allowed to create such relations:

- Rows - relation ends are the rows of a table. A link connected to the table is connected to a particular row. This mode is convenient for database diagrams.
- **Table** - relations target tables as whole entities, and not any particular row. This mode is convenient for class diagrams.
- **Both** - a mixed mode where links can target either a row of a table or the table itself.

Programmatically, links can be connected to a table in any way, regardless of the table's **ConnectionStyle**. The value of **ConnectionStyle** is considered only when end-users draw links to or from a table. To link to a table as an entity and not to a specific row, pass -1 as row argument of the CreateDiagramLink method.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.TableCustomDraw Property

Gets or sets the default type of custom painting done on tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CustomDraw TableCustomDraw { get; set; }
```

Visual Basic Copy Code

```
Public Property TableCustomDraw As CustomDraw
```

Property Value

A member of the CustomDraw enumeration. The default is **None**.

Remarks

The CustomDraw member of tables is initialized from this property.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
CellCustomDraw Property

Diagram.TableHandlesStyle Property

Gets or sets the default value for the **HandlesStyle** property of new tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HandlesStyle TableHandlesStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property TableHandlesStyle As HandlesStyle
```

Property Value

One of the HandlesStyle enumeration values.

Remarks

Default value for the **HandlesStyle** property of new tables, created either programmatically or by the user. That property specifies how selected tables are indicated visually, and how they can be moved, resized or connected to other objects.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.TableNodeStyle Property

Gets the theme-level style for table nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNodeStyle TableNodeStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property TableNodeStyle As TableNodeStyle
```

Property Value

See Also

Diagram.TablePen Property

Gets or sets the default value for the Pen property of tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen TablePen { get; set; }
```

Visual Basic Copy Code

```
Public Property TablePen As Pen
```

Property Value

An instance of the MindFusion.Drawing.Pen class.

Remarks

The property provides a default value for the pen, with which a table's frame is painted. The Pen property of new tables is initialized with a clone of **TablePen**, but after table creation the properties are independent - changing one of them does not change the other.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Brush and Pen Classes
TableBrush Property

Diagram.TableRowCount Property

Gets or sets the default number of table rows.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int TableRowCount { get; set; }
```

Visual Basic Copy Code

```
Public Property TableRowCount As Integer
```

Property Value

An integer value specifying the number of rows.

Remarks

Use the property to set the initial number of rows for newly created tables. Further customization of table nodes can be done through TableNode class properties and methods.

See Also

Diagram.TableRowHeight Property

Gets or sets the default height of tables' rows.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double TableRowHeight { get; set; }
```

Visual Basic Copy Code

```
Public Property TableRowHeight As Double
```

Property Value

A **double** value specifying the height of table rows. The default is 5.

Remarks

TableRowHeight is used to initialize the RowHeight property of new tables.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
TableColumnWidth Property

Diagram.TableShape Property

Gets or sets the default value of the Shape property of new tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SimpleShape TableShape { get; set; }
```

Visual Basic Copy Code

```
Public Property TableShape As SimpleShape
```

Property Value

A member of the SimpleShape enumeration. The default value is **Rectangle**.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.TablesScrollable Property

Gets or sets the default value for the Scrollable property of new tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool TablesScrollable { get; set; }
```

Visual Basic Copy Code

```
Public Property TablesScrollable As Boolean
```

Property Value

true if the scrolling is possible; otherwise, **false**. The default is **false**.

Remarks

The Scrollable property of each new table is initialized with the value of **TablesScrollable**. Small up/down scroll buttons appear in the caption of scrollable tables. With the help of the buttons users can scroll table rows.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
TableRowCount Property
TableNode.Scrollable Property

Diagram.Tag Property

Gets or sets custom data associated with this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Tag { get; set; }
```

Visual Basic Copy Code

```
Public Property Tag As Object
```

Property Value

An instance of any class derived from **System.Object**.

Remarks

Any type derived from the **Object** class can be assigned to this property. If the **Tag** property is set through the Windows Forms designer, only text can be assigned. A common use for the **Tag** property is to store data that is closely associated with diagrams and often must be accessed when processing a diagram instance.

If the tag is an instance of a value type, it will be serialized together with the diagram when saving or loading the diagram document. Reference types are serialized if marked with the **[Serializable]** attribute and implement the **ISerializable** interface.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.TextColor Property

Gets or sets the default color of text displayed inside diagram items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Color TextColor { get; set; }
```

Visual Basic Copy Code

```
Public Property TextColor As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** instance. The default is **Black**.

Remarks

Use the property to specify the default color of the text of diagram items.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
TextFormat Property

Diagram.TextFormat Property

Gets or sets the default alignment and formatting style for text displayed inside nodes and table cells.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringFormat TextFormat { get; set; }
```

Visual Basic Copy Code

```
Public Property TextFormat As StringFormat
```

Property Value

An instance of the StringFormat class. The default value for both **Alignment** and **LineAlignment** properties is **StringAlignment.Center**.

Remarks

When a new node is created, its TextFormat is initialized with the value of this property. The same applies for the TextFormat of the cells of a new table.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
TextColor Property

Diagram.Theme Property

Gets or sets the theme associated with the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Theme Theme { get; set; }
```

Visual Basic Copy Code

```
Public Property Theme As Theme
```

Property Value

An instance of the Theme class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.ToolTip Property

Gets or sets a short text message displayed when the mouse pointer rests over the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ToolTip { get; set; }
```

Visual Basic Copy Code

```
Public Property ToolTip As String
```

Property Value

The tool-tip text displayed for this diagram. The default is an empty string.

Remarks

This tool-tip is displayed when the mouse rests over an unoccupied document area. If the mouse points to an item, the ToolTip of that item will appear. Tool-tips can be enabled or disabled using the **ShowToolTips** property.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
DiagramView.ToolTipDelay Property

Diagram.TouchHitDistance Property

Specifies distance from adjustment handles at which they are still considered as hit when processing touch input.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Nullable<Double> TouchHitDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property TouchHitDistance As Nullable(Of Double)
```

Property Value

An optional **double** value specifying max touch distance from adjustment handles.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.TouchThreshold Property

Gets or sets the threshold value for easier touch interaction through adjustment handles.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public float TouchThreshold { get; set; }
```

Visual Basic Copy Code

```
Public Property TouchThreshold As Single
```

Property Value

A **float** value added to the hit-test area of adjustment handles. The default value is 0.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.TreeViewNodeStyle Property

Gets the theme-level style for tree view nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewNodeStyle TreeViewNodeStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property TreeViewNodeStyle As TreeViewNodeStyle
```

Property Value

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram.UndoManager Property

Gets a reference to the diagram's UndoManager instance.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public UndoManager UndoManager { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property UndoManager As UndoManager
```

Property Value

A reference to UndoManager instance.

Remarks

Use this property to control the undo manager operations, Undo or Redo commands, and access the command History.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace
Undo Manager

Diagram.ValidityChecks Property

Gets or sets a value indicating whether validity checks are performed each time an item is added to the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ValidityChecks { get; set; }
```

Visual Basic Copy Code

```
Public Property ValidityChecks As Boolean
```

Property Value

true to perform validity checks; otherwise, **false**. The default is **true**.

Remarks

If enabled, validity checks are performed each time an item is added to the diagram. That involves enumerating the Items collection and can slow up the process considerably for large diagrams. You can disable this property in order to skip the validity checks, however be sure that you don't add an item twice to the diagram and that links are created only between items in the same diagram.

See Also

Diagram.VisioNodeStyle Property

Gets the theme-level style for Visio nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public VisioNodeStyle VisioNodeStyle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property VisioNodeStyle As VisioNodeStyle
```

Property Value

An instance of the **VisioNodeStyle** class.

See Also

Diagram Members
Diagram Class
MindFusion.Diagramming Namespace

Diagram Events

DiagramBase Class

The base class of the Diagram and DiagramDocument classes.

Syntax

C# Copy Code

```
public abstract class DiagramBase : Component, INotifyPropertyChanged
```

Visual Basic Copy Code

```
Public MustInherit Class DiagramBase
    Inherits Component
    Implements INotifyPropertyChanged
```

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.DiagramBase
        MindFusion.Diagramming.Diagram
        MindFusion.Diagramming.DiagramDocument
        MindFusion.Diagramming.Diagram
        MindFusion.Diagramming.DiagramDocument

See Also

DiagramBase Members

The base class of the Diagram and DiagramDocument classes.

The following tables list the members exposed by the DiagramBase type.

Protected Constructors

	Name	Description
	DiagramBase	Initializes a new instance of the **DiagramBase** class.

Public Properties

	Name	Description
	XmlOptions	An XmlOptions object providing properties for customizing the XML serialization process.

Public Methods

	Name	Description
	RaiseNodeModifying	Raises the NodeModifying event.
	TrackMouseHover	Specifies whether the diagram view should track mouse hover events.

Protected Methods

	Name	Description
	OnActionRecorded	Raises the ActionRecorded event.
	OnActionRecording	Raises the ActionRecording event.
	OnActionRedone	Raises the ActionRedone event.
	OnActionUndone	Raises the ActionUndone event.
	OnBeginLoad	Raises the BeginLoad event.
	OnBoundsChanged	Raises the BoundsChanged event.
	OnCellClicked	Raises the CellClicked event.
	OnCellDoubleClicked	Raises the CellDoubleClicked event.
	OnCellTextEdited	Raises the CellTextEdited event.
	OnCellTextEditing	Raises the CellTextEditing event to validate entering inplace edit mode.
	OnClicked	Raises the Clicked event.
	OnContainerChildAdded	Raises the ContainerChildAdded event.
	OnContainerChildRemoved	Raises the ContainerChildRemoved event.
	OnContainerFolded	Raises the ContainerFolded event.
	OnContainerPageChanged	Raises the ContainerPageChanged event.
	OnContainerUnfolded	Raises the ContainerUnfolded event.
	OnContentVisibilityChanged	Raises the ContentVisibilityChanged event.

	OnDefaultShapeChanged	Raises the DefaultShapeChanged event.
	OnDeserializeTag	Raises the DeserializeTag event.
	OnDestinationAnchorChanged	Raises the DestinationAnchorChanged event.
	OnDirtyChanged	Raises the DirtyChanged event.
	OnDoubleClicked	Raises the DoubleClicked event.
	OnDrawAdjustmentHandles	Raises the DrawAdjustmentHandles event.
	OnDrawAnchorPoint	Raises the DrawAnchorPoint event.
	OnDrawBackground	Raises the DrawBackground event.
	OnDrawBranchIndicator	Raises the DrawBranchIndicator event.
	OnDrawCell	Raises the DrawCell event.
	OnDrawForeground	Raises the DrawForeground event.
	OnDrawLink	Raises the DrawLink event.
	OnDrawNode	Raises the DrawNode event.
	OnEndLoad	Raises the EndLoad event.
	OnExpandButtonClicked	Raises the ExpandButtonClicked event.
	OnGroupDestroyed	Raises the GroupDestroyed event.
	OnHitTestAdjustmentHandles	Raises the HitTestAdjustmentHandles event.
	OnInitializeLasso	Raises the InitializeLasso event.
	OnInitializeLink	Raises the InitializeLink event.
	OnInitializeNode	Raises the InitializeNode event.
	OnItemAdded	Raises the ItemAdded event.
	OnItemRemoved	Raises the ItemRemoved event.
	OnJsonDeserializeTag	Raises the JsonDeserializeTag event.
	OnJsonSerializeTag	Raises the JsonSerializeTag event.
	OnLayerVisibilityChanged	Raises the LayerVisibilityChanged event.
	OnLinkActivated	Raises the LinkActivated event.
	OnLinkClicked	Raises the LinkClicked event.
	OnLinkCreateCancelled	Raises the LinkCreateCancelled event.
	OnLinkCreated	Raises the LinkCreated event.
	OnLinkCreating	Raises the LinkCreating event.
	OnLinkDeactivated	Raises the LinkDeactivated event.
	OnLinkDeleted	Raises the LinkDeleted event.
	OnLinkDeleting	Raises the LinkDeleting event.
	OnLinkDeselected	Raises the LinkDeselected event.
	OnLinkDoubleClicked	Raises the LinkDoubleClicked event.
	OnLinkHovered	Raises the LinkHovered event.
	OnLinkModified	Raises the LinkModified event.

	OnLinkModifyCancelled	Raises the LinkModifyCancelled event.
	OnLinkModifying	Raises the LinkModifying event.
	OnLinkPasted	Raises the LinkPasted event.
	OnLinkRouted	Raises the LinkRouted event.
	OnLinkSelected	Raises the LinkSelected event.
	OnLinkSelecting	Raises the LinkSelecting event.
	OnLinkSplit	Raises the LinkSplit event.
	OnLinkStartModifying	Raises the LinkStartModifying event.
	OnLinkTextEdited	Raises the LinkTextEdited event.
	OnLinkTextEditing	Raises the LinkTextEditing event to validate entering inplace edit mode.
	OnMeasureUnitChanged	Raises the MeasureUnitChanged event.
	OnNodeActivated	Raises the NodeActivated event.
	OnNodeClicked	Raises the NodeClicked event.
	OnNodeCreateCancelled	Raises the NodeCreateCancelled event.
	OnNodeCreated	Raises the NodeCreated event.
	OnNodeCreating	Raises the NodeCreating event.
	OnNodeDeactivated	Raises the NodeDeactivated event.
	OnNodeDeleted	Raises the NodeDeleted event.
	OnNodeDeleting	Raises the NodeDeleting event.
	OnNodeDeselected	Raises the NodeDeselected event.
	OnNodeDoubleClicked	Raises the NodeDoubleClicked event.
	OnNodeHovered	Raises the NodeHovered event.
	OnNodeModified	Raises the NodeModified event.
	OnNodeModifyCancelled	Raises the NodeModifyCancelled event.
	OnNodeModifying	Raises the NodeModifying event.
	OnNodePasted	Raises the NodePasted event.
	OnNodeSelected	Raises the NodeSelected event.
	OnNodeSelecting	Raises the NodeSelecting event.
	OnNodeStartModifying	Raises the NodeStartModifying event.
	OnNodeTextEdited	Raises the NodeTextEdited event.
	OnNodeTextEditing	Raises the NodeTextEditing event to validate entering inplace edit mode.
	OnOriginAnchorChanged	Raises the OriginAnchorChanged event.
	OnPropertyChanged	Raises the PropertyChanged event.
	OnQueryAlignTarget	Raises the QueryAlignTarget event.
	OnRepaint	Raises the Repaint event.
	OnRequestClipboard	Raises the RequestClipboard event.
	OnRequestDrawInteractions	Raises the RequestDrawInteractions event.

	OnRequestTimer	Raises the RequestTimer event.
	OnSelectionChanged	Raises the SelectionChanged event.
	OnSelectionModified	Raises the SelectionModified event.
	OnSelectionModifying	Raises the SelectionModifying event.
	OnSerializeTag	Raises the SerializeTag event.
	OnSetSelfLoopShape	Raises the SetSelfLoopShape event.
	OnShapeModified	Raises the ShapeModified event.
	OnShapeNotDefined	Raises the ShapeNotDefined event.
	OnTableColumnResized	Raises the TableColumnResized event.
	OnTableColumnResizing	Raises the TableColumnResizing event.
	OnTableRowResized	Raises the TableRowResized event.
	OnTableRowResizing	Raises the TableRowResizing event.
	OnTableSectionCollapsed	Raises the TableSectionCollapsed event.
	OnTableSectionExpanded	Raises the TableSectionExpanded event.
	OnTreeCollapsed	Raises the TreeCollapsed event.
	OnTreeExpanded	Raises the TreeExpanded event.
	OnTreeItemTextEditing	Raises the TreeItemTextEditing event.
	OnValidateAnchorPoint	Raises the ValidateAnchorPoint event.
	OnValidateLinkCrossing	Raises the ValidateLinkCrossing event.
	OnViewRemoved	Raises the ViewRemoved event.

Public Events

	Name	Description
	ActionRecorded	Raised when an action is recorded in the undo buffer.
	ActionRecording	A validation event raised before an action is recorded in the undo buffer.
	ActionRedone	Raised when an action is redone.
	ActionUndone	Raised when an action is undone.
	ActiveLayerChanged	Raised when the diagram's active layer has changed.
	BeginLoad	Raised when the component starts loading diagram data from a file or memory stream.
	BoundsChanged	Raised when the Bounds property value has changed.
	CellClicked	Raised when the user clicks a table's cell.
	CellDoubleClicked	Raised when the user double-clicks a table's cell.
	CellTextEdited	Raised when the user edits the text of a table's cell.
	CellTextEditing	A validation event that lets you approve or reject changes made to the text of a table's cell.
	Clicked	Raised when the user clicks the diagram at a point where no items are located.

	ContainerChildAdded	Raised when a node is added to a ContainerNode, either programmatically or by the user.
	ContainerChildRemoved	Raised when a node is removed from a **ContainerNode**, either programmatically or by the user.
	ContainerFolded	Raised when the user folds a container by clicking the arrow-up button in the container's caption bar.
	ContainerPageChanged	Raised when the user changes current page of a PagedContainerNode.
	ContainerUnfolded	Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar.
	ContentVisibilityChanged	Raised when the visibility of container's content changes.
	DefaultShapeChanged	Raised when the DefaultShape property value has changed.
	DeserializeTag	Raised when the Tag of a diagram item must be deserialized from XML format.
	DestinationAnchorChanged	Raised when the DestinationAnchor property value of a link has changed.
	DirtyChanged	Raised when the **Dirty** property value has changed.
	DoubleClicked	Raised when the user double-clicks the diagram at a point where no items are located.
	DrawAdjustmentHandles	Raised when item's adjustment handles must be custom-drawn.
	DrawAnchorPoint	Raised when an anchor point must be custom-drawn.
	DrawBackground	Raised to let you custom draw the diagram background.
	DrawBranchIndicator	Raised when a branch indicator must be custom-drawn.
	DrawCell	Raised when a table's cell must be custom drawn.
	DrawForeground	Raised to let you custom draw the diagram foreground.
	DrawLink	Raised when a diagram link must be custom drawn.
	DrawNode	Raised when a diagram node must be custom drawn.
	EndLoad	Raised when the component ends loading diagram data from a file or memory stream.
	ExpandButtonClicked	Raised when the user clicks the [±] button of an Expandable node.
	GroupDestroyed	Raised when a Group object is destroyed.
	HitTestAdjustmentHandles	Raised when custom hit testing of item's adjustment handles must be performed.
	InitializeLasso	Raised when the user starts drawing a selection lasso, before it is rendered on screen.
	InitializeLink	Raised when the user starts drawing a new link, just after the DiagramLink instance is created.
	InitializeNode	Raised when the user starts drawing a new node, just after the DiagramNode instance is created.
	ItemAdded	Raised when an item is added to the diagram, either programmatically or interactively.
	ItemRemoved	Raised when an item is removed from the diagram, either programmatically or interactively.
	JsonDeserializeTag	Raised when the **Tag** of a diagram item must be deserialized from JSON.
	JsonSerializeTag	Raised when the **Tag** of a diagram item must be serialized to JSON.
	LayerVisibilityChanged	Raised when the visibility of a Layer changes.

	LinkActivated	Raised when a link becomes the ActiveItem.
	LinkClicked	Raised when the user clicks a link.
	LinkCreateCancelled	Raised when the user cancels link creation.
	LinkCreated	Raised when the user draws a new link.
	LinkCreating	A validation event raised while the user is drawing a new link.
	LinkDeactivated	Raised when a link is no longer the **ActiveItem**.
	LinkDeleted	Raised when a link is deleted, either programmatically or by the user.
	LinkDeleting	Raised when the user tries to delete a link, this event lets you cancel the operation.
	LinkDeselected	Raised when a link is deselected, either programmatically or by the user.
	LinkDoubleClicked	Raised when the user double-clicks a link.
	LinkHovered	Raised when the mouse pointer hovers over a link.
	LinkModified	Raised when the user moves any control point of a link.
	LinkModifyCancelled	Raised when the user cancels move or resize operation being done on a link.
	LinkModifying	A validation event raised while the user is modifying a link.
	LinkPasted	Raised when a link has been pasted from the clipboard.
	LinkRouted	Raised when a link is routed by the auto-routing algorithm.
	LinkSelected	Raised when a link is selected, either programmatically or by the user.
	LinkSelecting	Raised when the user tries to select a link, this event lets you cancel the operation.
	LinkSplit	Raised when the user splits a link interactively.
	LinkStartModifying	A validation event raised when the user starts modifying a link.
	LinkTextEdited	Raised when the user edits the text of a link.
	LinkTextEditing	A validation event that lets you approve or reject changes made to the text of a link.
	MeasureUnitChanged	Raised when the MeasureUnit property value has changed.
	NodeActivated	Raised when a node becomes the **ActiveItem**.
	NodeClicked	Raised when the user clicks a node.
	NodeCreateCancelled	Raised when the user cancels node creation.
	NodeCreated	Raised when the user draws a new node.
	NodeCreating	A validation event raised while the user is drawing a new node.
	NodeDeactivated	Raised when a node is no longer the **ActiveItem**.
	NodeDeleted	Raised when a node is deleted, either programmatically or by the user.
	NodeDeleting	Raised when the user tries to delete a node, this event lets you cancel the operation.
	NodeDeselected	Raised when a node is deselected, either programmatically or by the user.
	NodeDoubleClicked	Raised when the user double-clicks a node.
	NodeHovered	Raised when the mouse pointer hovers over a node.

	NodeModified	Raised when the user moves or resizes a node.
	NodeModifyCancelled	Raised when the user cancels move or resize operation being done on a node.
	NodeModifying	A validation event raised while the user is moving or resizing a node.
	NodePasted	Raised when a node has been pasted from the clipboard.
	NodeSelected	Raised when a node is selected, either programmatically or by the user.
	NodeSelecting	Raised when the user tries to select a node, this event lets you cancel the operation.
	NodeStartModifying	A validation event raised when the user starts modifying a node.
	NodeTextEdited	Raised when the user edits the text of a node.
	NodeTextEditing	A validation event that lets you approve or reject changes made to the text of a node.
	OriginAnchorChanged	Raised when the OriginAnchor property value of a link has changed.
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation.
	QueryAlignTarget	Raised to validate a node as target for alignment guides when AutoAlignNodes is enabled.
	Repaint	Raised when the diagram views should be repainted.
	RequestCacheReset	Raised when an object within the diagram requests a reset of the cached bitmap.
	RequestClipboard	Raised when an object within the diagram requests an operation with the clipboard.
	RequestDrawInteractions	For internal use.
	RequestTimer	Raised when an object within the diagram needs to start or stop a timer.
	SelectionChanged	Raised when an item is added to or removed from the selection
	SelectionModified	Raised when the user moves multiple selected items.
	SelectionModifying	Raised while the user is moving multiple selected items, this event lets you cancel the modification.
	SerializeTag	Raised when the **Tag** of a diagram item must be serialized into an XML document.
	SetSelfLoopShape	Raised when a link becomes a self-loop, giving you a chance to override the default look shape.
	ShapeModified	Raised when the user modifies the shape of a ShapeNode by moving one of the control points corresponding to shape formula parameters.
	ShapeNotDefined	Raised when the diagram being loaded contains nodes with an undefined shape.
	TableColumnResized	Raised when the user finishes resizing a column in a table.
	TableColumnResizing	Raised when the user resizes a column in a table interactively.
	TableRowResized	Raised when the user finishes resizing a row in a table.
	TableRowResizing	Raised when the user resizes a row in a table interactively.
	TableSectionCollapsed	Raised when the user collapses the rows of a table's section by clicking the [-] button in the section's header row.
	TableSectionExpanded	Raised when the user expands the rows of a table's section by clicking the [+] button in the section's header row.

	TreeCollapsed	Raised when a user collapses a tree branch by clicking the [-] button of an **Expandable** node.
	TreeExpanded	Raised when a user expands a tree branch by clicking the [+] button of an **Expandable** node.
	TreeItemTextEditing	A validation event that lets you approve or reject changes made to the text of a TreeViewNode's item.
	ValidateAnchorPoint	A validation event that lets you approve or reject connecting a link to an anchor point.
	ValidateLinkCrossing	A validation event that lets you approve or reject displaying a decoration at the intersection point of links.
	ViewRemoved	Raised when a DiagramView is detached from this diagram.

See Also

DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase Constructor

Initializes a new instance of the DiagramBase class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected DiagramBase ()
```

Visual Basic Copy Code

```
Protected Sub New ()
```

See Also

DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase Methods

DiagramBase.OnActionRecorded Method

Raises the ActionRecorded event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnActionRecorded (
    UndoEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnActionRecorded( _
    e As UndoEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnActionRecording Method

Raises the ActionRecording event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnActionRecording (
    UndoValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnActionRecording( _
    e As UndoValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnActionRedone Method

Raises the ActionRedone event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnActionRedone (
    UndoEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnActionRedone( _
    e As UndoEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnActionUndone Method

Raises the ActionUndone event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnActionUndone (
    UndoEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnActionUndone( _
    e As UndoEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnBeginLoad Method

Raises the BeginLoad event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnBeginLoad (
    EventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnBeginLoad( _
    e As EventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnBoundsChanged Method

Raises the BoundsChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnBoundsChanged (
    EventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnBoundsChanged( _
    e As EventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnCellClicked Method

Raises the CellClicked event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnCellClicked (
    CellEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnCellClicked( _
    e As CellEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnCellDoubleClicked Method

Raises the CellDoubleClicked event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnCellDoubleClicked (
    CellEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnCellDoubleClicked( _
    e As CellEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnCellTextEdited Method

Raises the CellTextEdited event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnCellTextEdited (
    EditCellTextEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnCellTextEdited( _
    e As EditCellTextEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnCellTextEditing Method

Raises the CellTextEditing event to validate entering inplace edit mode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnCellTextEditing (
    CellValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnCellTextEditing( _
    e As CellValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnClicked Method

Raises the Clicked event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnClicked (
    DiagramEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnClicked( _
    e As DiagramEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnContainerChildAdded Method

Raises the ContainerChildAdded event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnContainerChildAdded (
    ContainerChildEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnContainerChildAdded( _
    e As ContainerChildEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnContainerChildRemoved Method

Raises the ContainerChildRemoved event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnContainerChildRemoved (
    ContainerChildEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnContainerChildRemoved( _
    e As ContainerChildEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnContainerFolded Method

Raises the ContainerFolded event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnContainerFolded (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnContainerFolded( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnContainerPageChanged Method

Raises the ContainerPageChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnContainerPageChanged (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnContainerPageChanged( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnContainerUnfolded Method

Raises the ContainerUnfolded event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnContainerUnfolded (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnContainerUnfolded( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnContentVisibilityChanged Method

Raises the ContentVisibilityChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnContentVisibilityChanged (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnContentVisibilityChanged( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDefaultShapeChanged Method

Raises the DefaultShapeChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDefaultShapeChanged (
    EventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDefaultShapeChanged( _
    e As EventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDeserializeTag Method

Raises the DeserializeTag event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDeserializeTag (
    SerializeTagEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDeserializeTag( _
    e As SerializeTagEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDestinationAnchorChanged Method

Raises the DestinationAnchorChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDestinationAnchorChanged (
    AnchorPointEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDestinationAnchorChanged( _
    e As AnchorPointEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDirtyChanged Method

Raises the DirtyChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDirtyChanged (
    EventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDirtyChanged( _
    e As EventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDoubleClicked Method

Raises the DoubleClicked event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDoubleClicked (
    DiagramEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDoubleClicked( _
    e As DiagramEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDrawAdjustmentHandles Method

Raises the DrawAdjustmentHandles event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDrawAdjustmentHandles (
    DrawItemEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDrawAdjustmentHandles( _
    e As DrawItemEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDrawAnchorPoint Method

Raises the DrawAnchorPoint event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDrawAnchorPoint (
    DrawAnchorPointEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDrawAnchorPoint( _
    e As DrawAnchorPointEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDrawBackground Method

Raises the DrawBackground event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDrawBackground (
    DiagramEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDrawBackground( _
    e As DiagramEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDrawBranchIndicator Method

Raises the DrawBranchIndicator event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDrawBranchIndicator (
    DrawBranchIndicatorEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDrawBranchIndicator( _
    e As DrawBranchIndicatorEventArgs _
)
```

Parameters

e
A DrawBranchIndicatorEventArgs that contains the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDrawCell Method

Raises the DrawCell event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDrawCell (
    DrawCellEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDrawCell( _
    e As DrawCellEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDrawForeground Method

Raises the DrawForeground event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDrawForeground (
    DiagramEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDrawForeground( _
    e As DiagramEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDrawLink Method

Raises the DrawLink event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDrawLink (
    DrawLinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDrawLink( _
    e As DrawLinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnDrawNode Method

Raises the DrawNode event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDrawNode (
    DrawNodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDrawNode( _
    e As DrawNodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnEndLoad Method

Raises the EndLoad event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnEndLoad (
    EventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnEndLoad( _
    e As EventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnExpandButtonClicked Method

Raises the ExpandButtonClicked event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnExpandButtonClicked (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnExpandButtonClicked( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnGroupDestroyed Method

Raises the GroupDestroyed event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnGroupDestroyed (
    GroupEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnGroupDestroyed( _
    e As GroupEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnHitTestAdjustmentHandles Method

Raises the HitTestAdjustmentHandles event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnHitTestAdjustmentHandles (
    HitTestEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnHitTestAdjustmentHandles( _
    e As HitTestEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnInitializeLasso Method

Raises the InitializeLasso event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnInitializeLasso (
    SelectionEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnInitializeLasso( _
    e As SelectionEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnInitializeLink Method

Raises the InitializeLink event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnInitializeLink (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnInitializeLink( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnInitializeNode Method

Raises the InitializeNode event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnInitializeNode (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnInitializeNode( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnItemAdded Method

Raises the ItemAdded event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnItemAdded (
    ItemEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnItemAdded( _
    e As ItemEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnItemRemoved Method

Raises the ItemRemoved event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnItemRemoved (
    ItemEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnItemRemoved( _
    e As ItemEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnJsonDeserializeTag Method

Raises the JsonDeserializeTag event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnJsonDeserializeTag (
    JsonSerializeTagEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnJsonDeserializeTag( _
    e As JsonSerializeTagEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnJsonSerializeTag Method

Raises the JsonSerializeTag event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnJsonSerializeTag (
    JsonSerializeTagEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnJsonSerializeTag( _
    e As JsonSerializeTagEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLayerVisibilityChanged Method

Raises the LayerVisibilityChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLayerVisibilityChanged (
    LayerEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLayerVisibilityChanged( _
    e As LayerEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkActivated Method

Raises the LinkActivated event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkActivated (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkActivated( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkClicked Method

Raises the LinkClicked event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkClicked (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkClicked( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkCreateCancelled Method

Raises the LinkCreateCancelled event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkCreateCancelled (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkCreateCancelled( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkCreated Method

Raises the LinkCreated event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkCreated (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkCreated( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkCreating Method

Raises the LinkCreating event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkCreating (
    LinkValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkCreating( _
    e As LinkValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkDeactivated Method

Raises the LinkDeactivated event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkDeactivated (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkDeactivated( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkDeleted Method

Raises the LinkDeleted event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkDeleted (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkDeleted( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkDeleting Method

Raises the LinkDeleting event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkDeleting (
    LinkValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkDeleting( _
    e As LinkValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkDeselected Method

Raises the LinkDeselected event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkDeselected (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkDeselected( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkDoubleClicked Method

Raises the LinkDoubleClicked event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkDoubleClicked (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkDoubleClicked( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkHovered Method

Raises the LinkHovered event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkHovered (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkHovered( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkModified Method

Raises the LinkModified event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkModified (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkModified( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkModifyCancelled Method

Raises the LinkModifyCancelled event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkModifyCancelled (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkModifyCancelled( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkModifying Method

Raises the LinkModifying event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkModifying (
    LinkValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkModifying( _
    e As LinkValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkPasted Method

Raises the LinkPasted event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkPasted (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkPasted( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkRouted Method

Raises the LinkRouted event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkRouted (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkRouted( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkSelected Method

Raises the LinkSelected event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkSelected (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkSelected( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkSelecting Method

Raises the LinkSelecting event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkSelecting (
    LinkValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkSelecting( _
    e As LinkValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkSplit Method

Raises the LinkSplit event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkSplit (
    LinkEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkSplit( _
    e As LinkEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkStartModifying Method

Raises the LinkStartModifying event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkStartModifying (
    LinkValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkStartModifying( _
    e As LinkValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkTextEdited Method

Raises the LinkTextEdited event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkTextEdited (
    EditLinkTextEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkTextEdited( _
    e As EditLinkTextEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnLinkTextEditing Method

Raises the LinkTextEditing event to validate entering inplace edit mode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnLinkTextEditing (
    LinkValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnLinkTextEditing( _
    e As LinkValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnMeasureUnitChanged Method

Raises the MeasureUnitChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnMeasureUnitChanged (
    EventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnMeasureUnitChanged( _
    e As EventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeActivated Method

Raises the NodeActivated event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeActivated (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeActivated( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeClicked Method

Raises the NodeClicked event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeClicked (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeClicked( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeCreateCancelled Method

Raises the NodeCreateCancelled event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeCreateCancelled (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeCreateCancelled( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeCreated Method

Raises the NodeCreated event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeCreated (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeCreated( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeCreating Method

Raises the NodeCreating event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeCreating (
    NodeValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeCreating( _
    e As NodeValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeDeactivated Method

Raises the NodeDeactivated event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeDeactivated (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeDeactivated( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeDeleted Method

Raises the NodeDeleted event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeDeleted (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeDeleted( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeDeleting Method

Raises the NodeDeleting event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeDeleting (
    NodeValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeDeleting( _
    e As NodeValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeDeselected Method

Raises the NodeDeselected event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeDeselected (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeDeselected( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeDoubleClicked Method

Raises the NodeDoubleClicked event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeDoubleClicked (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeDoubleClicked( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeHovered Method

Raises the NodeHovered event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeHovered (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeHovered( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeModified Method

Raises the NodeModified event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeModified (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeModified( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeModifyCancelled Method

Raises the NodeModifyCancelled event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeModifyCancelled (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeModifyCancelled( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeModifying Method

Raises the NodeModifying event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeModifying (
    NodeValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeModifying( _
    e As NodeValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodePasted Method

Raises the NodePasted event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodePasted (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodePasted( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeSelected Method

Raises the NodeSelected event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeSelected (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeSelected( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeSelecting Method

Raises the NodeSelecting event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeSelecting (
    NodeValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeSelecting( _
    e As NodeValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeStartModifying Method

Raises the NodeStartModifying event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeStartModifying (
    NodeValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeStartModifying( _
    e As NodeValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeTextEdited Method

Raises the NodeTextEdited event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeTextEdited (
    EditNodeTextEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeTextEdited( _
    e As EditNodeTextEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnNodeTextEditing Method

Raises the NodeTextEditing event to validate entering inplace edit mode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnNodeTextEditing (
    NodeValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnNodeTextEditing( _
    e As NodeValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnOriginAnchorChanged Method

Raises the OriginAnchorChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnOriginAnchorChanged (
    AnchorPointEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnOriginAnchorChanged( _
    e As AnchorPointEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnPropertyChanged Method

Raises the PropertyChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnPropertyChanged (
    PropertyChangedEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnPropertyChanged( _
    e As PropertyChangedEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnQueryAlignTarget Method

Raises the QueryAlignTarget event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnQueryAlignTarget (
    NodeValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnQueryAlignTarget( _
    e As NodeValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnRepaint Method

Raises the Repaint event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnRepaint (
    RepaintEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnRepaint( _
    e As RepaintEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnRequestClipboard Method

Raises the RequestClipboard event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnRequestClipboard (
    ClipboardEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnRequestClipboard( _
    e As ClipboardEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnRequestDrawInteractions Method

Raises the RequestDrawInteractions event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnRequestDrawInteractions (
    DiagramEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnRequestDrawInteractions( _
    e As DiagramEventArgs _
)
```

Parameters

e
DiagramEventArgs instance.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnRequestTimer Method

Raises the RequestTimer event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnRequestTimer (
    TimerEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnRequestTimer( _
    e As TimerEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnSelectionChanged Method

Raises the SelectionChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnSelectionChanged (
    EventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnSelectionChanged( _
    e As EventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnSelectionModified Method

Raises the SelectionModified event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnSelectionModified (
    SelectionEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnSelectionModified( _
    e As SelectionEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnSelectionModifying Method

Raises the SelectionModifying event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnSelectionModifying (
    SelectionValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnSelectionModifying( _
    e As SelectionValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnSerializeTag Method

Raises the SerializeTag event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnSerializeTag (
    SerializeTagEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnSerializeTag( _
    e As SerializeTagEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnSetSelfLoopShape Method

Raises the SetSelfLoopShape event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnSetSelfLoopShape (
    LinkShapeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnSetSelfLoopShape( _
    e As LinkShapeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnShapeModified Method

Raises the ShapeModified event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnShapeModified (
    ControlPointEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnShapeModified( _
    e As ControlPointEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnShapeNotDefined Method

Raises the ShapeNotDefined event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnShapeNotDefined (
    ShapeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnShapeNotDefined( _
    e As ShapeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnTableColumnResized Method

Raises the TableColumnResized event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnTableColumnResized (
    CellEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnTableColumnResized( _
    e As CellEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnTableColumnResizing Method

Raises the TableColumnResizing event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnTableColumnResizing (
    CellValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnTableColumnResizing( _
    e As CellValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnTableRowResized Method

Raises the TableRowResized event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnTableRowResized (
    CellEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnTableRowResized( _
    e As CellEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnTableRowResizing Method

Raises the TableRowResizing event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnTableRowResizing (
    CellValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnTableRowResizing( _
    e As CellValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnTableSectionCollapsed Method

Raises the TableSectionCollapsed event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnTableSectionCollapsed (
    RowEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnTableSectionCollapsed( _
    e As RowEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnTableSectionExpanded Method

Raises the TableSectionExpanded event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnTableSectionExpanded (
    RowEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnTableSectionExpanded( _
    e As RowEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnTreeCollapsed Method

Raises the TreeCollapsed event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnTreeCollapsed (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnTreeCollapsed( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnTreeExpanded Method

Raises the TreeExpanded event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnTreeExpanded (
    NodeEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnTreeExpanded( _
    e As NodeEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnTreeItemTextEditing Method

Raises the TreeItemTextEditing event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnTreeItemTextEditing (
    TreeItemValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnTreeItemTextEditing( _
    e As TreeItemValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnValidateAnchorPoint Method

Raises the ValidateAnchorPoint event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnValidateAnchorPoint (
    LinkValidationEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnValidateAnchorPoint( _
    e As LinkValidationEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnValidateLinkCrossing Method

Raises the ValidateLinkCrossing event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnValidateLinkCrossing (
    LinkCrossingEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnValidateLinkCrossing( _
    e As LinkCrossingEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.OnViewRemoved Method

Raises the ViewRemoved event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnViewRemoved (
    CancelEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnViewRemoved( _
    e As CancelEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.RaiseNodeModifying Method

Raises the NodeModifying event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RaiseNodeModifying (
    NodeValidationEventArgs e
)
```

Visual Basic Copy Code

```
Public Sub RaiseNodeModifying( _
    e As NodeValidationEventArgs _
)
```

Parameters

e
NodeValidationEventArgs instance.

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase.TrackMouseHover Method

Specifies whether the diagram view should track mouse hover events.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool TrackMouseHover ()
```

Visual Basic Copy Code

```
Public Overridable Function TrackMouseHover () As Boolean
```

Return Value

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase Properties

DiagramBase.XmlOptions Property

An XmlOptions object providing properties for customizing the XML serialization process.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlOptions XmlOptions { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property XmlOptions As XmlOptions
```

Property Value

See Also

DiagramBase Members
DiagramBase Class
MindFusion.Diagramming Namespace

DiagramBase Events

DiagramDocument Class

Represents a collection of diagram pages.

Syntax

C# Copy Code

```
public class DiagramDocument : DiagramBase
```

Visual Basic Copy Code

```
Public Class DiagramDocument
    Inherits DiagramBase
```

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.DiagramBase
        MindFusion.Diagramming.DiagramDocument

See Also

DiagramDocument Members

Represents a collection of diagram pages.

The following tables list the members exposed by the DiagramDocument type.

Public Constructors

	Name	Description
	DiagramDocument	Initializes a new instance of the **DiagramDocument** class.

Public Properties

	Name	Description
	Dirty	Gets a value indicating whether the document has changed since the last save.
	Pages	Gets a list of the diagram pages contained in this document.
	XmlOptions	An XmlOptions object providing properties for customizing the XML serialization process. (Inherited from DiagramBase.)

Public Methods

	Name	Description
	Clone	Creates a copy of this DiagramDocument.
	CopyTo	Copies this document's pages to specified target document.
	LoadFromJson	Loads the diagram document from a JSON-formatted string.
	LoadFromJsonAsync	Loads the diagram document from a JSON-formatted string.
	LoadFromJsonFile	Loads the diagram document from a JSON file.
	LoadFromJsonFileAsync	Loads the diagram document from a JSON file.
	LoadFromJsonObject	Loads the diagram document from a JSON object.
	LoadFromString	Loads the document from the specified string.
	LoadFromXml	Overloaded. Loads the document from an XML document or file.
	LoadFromXmlAsync	Overloaded.
	RaiseNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	SaveToJson	Saves the diagram document to a JSON-formatted string.
	SaveToJsonAsync	Saves the diagram document to a JSON-formatted string.
	SaveToJsonFile	Saves the diagram document into a JSON file.
	SaveToJsonFileAsync	Saves the diagram document into a JSON file.
	SaveToJsonObject	Saves the diagram document to a JSON object.
	SaveToString	Saves the document as an ASCII string.
	SaveToXml	Overloaded. Saves the document to an XML document or file.
	SaveToXmlAsync	Overloaded.

	TrackMouseHover	Specifies whether the diagram view should track mouse hover events. (Inherited from DiagramBase.)

Protected Methods

	Name	Description
	OnActionRecorded	Raises the ActionRecorded event. (Inherited from DiagramBase.)
	OnActionRecording	Raises the ActionRecording event. (Inherited from DiagramBase.)
	OnActionRedone	Raises the ActionRedone event. (Inherited from DiagramBase.)
	OnActionUndone	Raises the ActionUndone event. (Inherited from DiagramBase.)
	OnBeginLoad	Raises the BeginLoad event. (Inherited from DiagramBase.)
	OnBoundsChanged	Raises the BoundsChanged event. (Inherited from DiagramBase.)
	OnCellClicked	Raises the CellClicked event. (Inherited from DiagramBase.)
	OnCellDoubleClicked	Raises the CellDoubleClicked event. (Inherited from DiagramBase.)
	OnCellTextEdited	Raises the CellTextEdited event. (Inherited from DiagramBase.)
	OnCellTextEditing	Raises the CellTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnClicked	Raises the Clicked event. (Inherited from DiagramBase.)
	OnContainerChildAdded	Raises the ContainerChildAdded event. (Inherited from DiagramBase.)
	OnContainerChildRemoved	Raises the ContainerChildRemoved event. (Inherited from DiagramBase.)
	OnContainerFolded	Raises the ContainerFolded event. (Inherited from DiagramBase.)
	OnContainerPageChanged	Raises the ContainerPageChanged event. (Inherited from DiagramBase.)
	OnContainerUnfolded	Raises the ContainerUnfolded event. (Inherited from DiagramBase.)
	OnContentVisibilityChanged	Raises the ContentVisibilityChanged event. (Inherited from DiagramBase.)
	OnDefaultShapeChanged	Raises the DefaultShapeChanged event. (Inherited from DiagramBase.)
	OnDeserializeTag	Raises the DeserializeTag event. (Inherited from DiagramBase.)
	OnDestinationAnchorChanged	Raises the DestinationAnchorChanged event. (Inherited from DiagramBase.)
	OnDirtyChanged	Raises the DirtyChanged event. (Inherited from DiagramBase.)
	OnDoubleClicked	Raises the DoubleClicked event. (Inherited from DiagramBase.)
	OnDrawAdjustmentHandles	Raises the DrawAdjustmentHandles event. (Inherited from DiagramBase.)
	OnDrawAnchorPoint	Raises the DrawAnchorPoint event. (Inherited from DiagramBase.)
	OnDrawBackground	Raises the DrawBackground event. (Inherited from DiagramBase.)
	OnDrawBranchIndicator	Raises the DrawBranchIndicator event. (Inherited from DiagramBase.)
	OnDrawCell	Raises the DrawCell event. (Inherited from DiagramBase.)
	OnDrawForeground	Raises the DrawForeground event. (Inherited from DiagramBase.)
	OnDrawLink	Raises the DrawLink event. (Inherited from DiagramBase.)
	OnDrawNode	Raises the DrawNode event. (Inherited from DiagramBase.)
	OnEndLoad	Raises the EndLoad event. (Inherited from DiagramBase.)
	OnExpandButtonClicked	Raises the ExpandButtonClicked event. (Inherited from DiagramBase.)

	OnGroupDestroyed	Raises the GroupDestroyed event. (Inherited from DiagramBase.)
	OnHitTestAdjustmentHandles	Raises the HitTestAdjustmentHandles event. (Inherited from DiagramBase.)
	OnInitializeLasso	Raises the InitializeLasso event. (Inherited from DiagramBase.)
	OnInitializeLink	Raises the InitializeLink event. (Inherited from DiagramBase.)
	OnInitializeNode	Raises the InitializeNode event. (Inherited from DiagramBase.)
	OnItemAdded	Raises the ItemAdded event. (Inherited from DiagramBase.)
	OnItemRemoved	Raises the ItemRemoved event. (Inherited from DiagramBase.)
	OnJsonDeserializeTag	Raises the JsonDeserializeTag event. (Inherited from DiagramBase.)
	OnJsonSerializeTag	Raises the JsonSerializeTag event. (Inherited from DiagramBase.)
	OnLayerVisibilityChanged	Raises the LayerVisibilityChanged event. (Inherited from DiagramBase.)
	OnLinkActivated	Raises the LinkActivated event. (Inherited from DiagramBase.)
	OnLinkClicked	Raises the LinkClicked event. (Inherited from DiagramBase.)
	OnLinkCreateCancelled	Raises the LinkCreateCancelled event. (Inherited from DiagramBase.)
	OnLinkCreated	Raises the LinkCreated event. (Inherited from DiagramBase.)
	OnLinkCreating	Raises the LinkCreating event. (Inherited from DiagramBase.)
	OnLinkDeactivated	Raises the LinkDeactivated event. (Inherited from DiagramBase.)
	OnLinkDeleted	Raises the LinkDeleted event. (Inherited from DiagramBase.)
	OnLinkDeleting	Raises the LinkDeleting event. (Inherited from DiagramBase.)
	OnLinkDeselected	Raises the LinkDeselected event. (Inherited from DiagramBase.)
	OnLinkDoubleClicked	Raises the LinkDoubleClicked event. (Inherited from DiagramBase.)
	OnLinkHovered	Raises the LinkHovered event. (Inherited from DiagramBase.)
	OnLinkModified	Raises the LinkModified event. (Inherited from DiagramBase.)
	OnLinkModifyCancelled	Raises the LinkModifyCancelled event. (Inherited from DiagramBase.)
	OnLinkModifying	Raises the LinkModifying event. (Inherited from DiagramBase.)
	OnLinkPasted	Raises the LinkPasted event. (Inherited from DiagramBase.)
	OnLinkRouted	Raises the LinkRouted event. (Inherited from DiagramBase.)
	OnLinkSelected	Raises the LinkSelected event. (Inherited from DiagramBase.)
	OnLinkSelecting	Raises the LinkSelecting event. (Inherited from DiagramBase.)
	OnLinkSplit	Raises the LinkSplit event. (Inherited from DiagramBase.)
	OnLinkStartModifying	Raises the LinkStartModifying event. (Inherited from DiagramBase.)
	OnLinkTextEdited	Raises the LinkTextEdited event. (Inherited from DiagramBase.)
	OnLinkTextEditing	Raises the LinkTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnMeasureUnitChanged	Raises the MeasureUnitChanged event. (Inherited from DiagramBase.)
	OnNodeActivated	Raises the NodeActivated event. (Inherited from DiagramBase.)
	OnNodeClicked	Raises the NodeClicked event. (Inherited from DiagramBase.)
	OnNodeCreateCancelled	Raises the NodeCreateCancelled event. (Inherited from DiagramBase.)

	OnNodeCreated	Raises the NodeCreated event. (Inherited from DiagramBase.)
	OnNodeCreating	Raises the NodeCreating event. (Inherited from DiagramBase.)
	OnNodeDeactivated	Raises the NodeDeactivated event. (Inherited from DiagramBase.)
	OnNodeDeleted	Raises the NodeDeleted event. (Inherited from DiagramBase.)
	OnNodeDeleting	Raises the NodeDeleting event. (Inherited from DiagramBase.)
	OnNodeDeselected	Raises the NodeDeselected event. (Inherited from DiagramBase.)
	OnNodeDoubleClicked	Raises the NodeDoubleClicked event. (Inherited from DiagramBase.)
	OnNodeHovered	Raises the NodeHovered event. (Inherited from DiagramBase.)
	OnNodeModified	Raises the NodeModified event. (Inherited from DiagramBase.)
	OnNodeModifyCancelled	Raises the NodeModifyCancelled event. (Inherited from DiagramBase.)
	OnNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	OnNodePasted	Raises the NodePasted event. (Inherited from DiagramBase.)
	OnNodeSelected	Raises the NodeSelected event. (Inherited from DiagramBase.)
	OnNodeSelecting	Raises the NodeSelecting event. (Inherited from DiagramBase.)
	OnNodeStartModifying	Raises the NodeStartModifying event. (Inherited from DiagramBase.)
	OnNodeTextEdited	Raises the NodeTextEdited event. (Inherited from DiagramBase.)
	OnNodeTextEditing	Raises the NodeTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnOriginAnchorChanged	Raises the OriginAnchorChanged event. (Inherited from DiagramBase.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from DiagramBase.)
	OnQueryAlignTarget	Raises the QueryAlignTarget event. (Inherited from DiagramBase.)
	OnRepaint	Raises the Repaint event. (Inherited from DiagramBase.)
	OnRequestClipboard	Raises the RequestClipboard event. (Inherited from DiagramBase.)
	OnRequestDrawInteractions	Raises the RequestDrawInteractions event. (Inherited from DiagramBase.)
	OnRequestTimer	Raises the RequestTimer event. (Inherited from DiagramBase.)
	OnSelectionChanged	Raises the SelectionChanged event. (Inherited from DiagramBase.)
	OnSelectionModified	Raises the SelectionModified event. (Inherited from DiagramBase.)
	OnSelectionModifying	Raises the SelectionModifying event. (Inherited from DiagramBase.)
	OnSerializeTag	Raises the SerializeTag event. (Inherited from DiagramBase.)
	OnSetSelfLoopShape	Raises the SetSelfLoopShape event. (Inherited from DiagramBase.)
	OnShapeModified	Raises the ShapeModified event. (Inherited from DiagramBase.)
	OnShapeNotDefined	Raises the ShapeNotDefined event. (Inherited from DiagramBase.)
	OnTableColumnResized	Raises the TableColumnResized event. (Inherited from DiagramBase.)
	OnTableColumnResizing	Raises the TableColumnResizing event. (Inherited from DiagramBase.)
	OnTableRowResized	Raises the TableRowResized event. (Inherited from DiagramBase.)
	OnTableRowResizing	Raises the TableRowResizing event. (Inherited from DiagramBase.)
	OnTableSectionCollapsed	Raises the TableSectionCollapsed event. (Inherited from DiagramBase.)

	OnTableSectionExpanded	Raises the TableSectionExpanded event. (Inherited from DiagramBase.)
	OnTreeCollapsed	Raises the TreeCollapsed event. (Inherited from DiagramBase.)
	OnTreeExpanded	Raises the TreeExpanded event. (Inherited from DiagramBase.)
	OnTreeItemTextEditing	Raises the TreeItemTextEditing event. (Inherited from DiagramBase.)
	OnValidateAnchorPoint	Raises the ValidateAnchorPoint event. (Inherited from DiagramBase.)
	OnValidateLinkCrossing	Raises the ValidateLinkCrossing event. (Inherited from DiagramBase.)
	OnViewRemoved	Raises the ViewRemoved event. (Inherited from DiagramBase.)

Public Events

	Name	Description
	ActionRecorded	Raised when an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRecording	A validation event raised before an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRedone	Raised when an action is redone. (Inherited from DiagramBase.)
	ActionUndone	Raised when an action is undone. (Inherited from DiagramBase.)
	ActiveLayerChanged	Raised when the diagram's active layer has changed. (Inherited from DiagramBase.)
	BeginLoad	Raised when the component starts loading diagram data from a file or memory stream. (Inherited from DiagramBase.)
	BoundsChanged	Raised when the Bounds property value has changed. (Inherited from DiagramBase.)
	CellClicked	Raised when the user clicks a table's cell. (Inherited from DiagramBase.)
	CellDoubleClicked	Raised when the user double-clicks a table's cell. (Inherited from DiagramBase.)
	CellTextEdited	Raised when the user edits the text of a table's cell. (Inherited from DiagramBase.)
	CellTextEditing	A validation event that lets you approve or reject changes made to the text of a table's cell. (Inherited from DiagramBase.)
	Clicked	Raised when the user clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	ContainerChildAdded	Raised when a node is added to a ContainerNode, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerChildRemoved	Raised when a node is removed from a **ContainerNode**, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerFolded	Raised when the user folds a container by clicking the arrow-up button in the container's caption bar. (Inherited from DiagramBase.)
	ContainerPageChanged	Raised when the user changes current page of a PagedContainerNode. (Inherited from DiagramBase.)
	ContainerUnfolded	Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar. (Inherited from DiagramBase.)
	ContentVisibilityChanged	Raised when the visibility of container's content changes. (Inherited from DiagramBase.)
	DefaultShapeChanged	Raised when the DefaultShape property value has changed. (Inherited from DiagramBase.)

	DeserializeTag	Raised when the Tag of a diagram item must be deserialized from XML format. (Inherited from DiagramBase.)
	DestinationAnchorChanged	Raised when the DestinationAnchor property value of a link has changed. (Inherited from DiagramBase.)
	DirtyChanged	Raised when the **Dirty** property value has changed. (Inherited from DiagramBase.)
	DoubleClicked	Raised when the user double-clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	DrawAdjustmentHandles	Raised when item's adjustment handles must be custom-drawn. (Inherited from DiagramBase.)
	DrawAnchorPoint	Raised when an anchor point must be custom-drawn. (Inherited from DiagramBase.)
	DrawBackground	Raised to let you custom draw the diagram background. (Inherited from DiagramBase.)
	DrawBranchIndicator	Raised when a branch indicator must be custom-drawn. (Inherited from DiagramBase.)
	DrawCell	Raised when a table's cell must be custom drawn. (Inherited from DiagramBase.)
	DrawForeground	Raised to let you custom draw the diagram foreground. (Inherited from DiagramBase.)
	DrawLink	Raised when a diagram link must be custom drawn. (Inherited from DiagramBase.)
	DrawNode	Raised when a diagram node must be custom drawn. (Inherited from DiagramBase.)
	EndLoad	Raised when the component ends loading diagram data from a file or memory stream. (Inherited from DiagramBase.)
	ExpandButtonClicked	Raised when the user clicks the [±] button of an Expandable node. (Inherited from DiagramBase.)
	GroupDestroyed	Raised when a Group object is destroyed. (Inherited from DiagramBase.)
	HitTestAdjustmentHandles	Raised when custom hit testing of item's adjustment handles must be performed. (Inherited from DiagramBase.)
	InitializeLasso	Raised when the user starts drawing a selection lasso, before it is rendered on screen. (Inherited from DiagramBase.)
	InitializeLink	Raised when the user starts drawing a new link, just after the DiagramLink instance is created. (Inherited from DiagramBase.)
	InitializeNode	Raised when the user starts drawing a new node, just after the DiagramNode instance is created. (Inherited from DiagramBase.)
	ItemAdded	Raised when an item is added to the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	ItemRemoved	Raised when an item is removed from the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	JsonDeserializeTag	Raised when the **Tag** of a diagram item must be deserialized from JSON. (Inherited from DiagramBase.)
	JsonSerializeTag	Raised when the **Tag** of a diagram item must be serialized to JSON. (Inherited from DiagramBase.)
	LayerVisibilityChanged	Raised when the visibility of a Layer changes. (Inherited from DiagramBase.)
	LinkActivated	Raised when a link becomes the ActiveItem. (Inherited from DiagramBase.)
	LinkClicked	Raised when the user clicks a link. (Inherited from DiagramBase.)

	LinkCreateCancelled	Raised when the user cancels link creation. (Inherited from DiagramBase.)
	LinkCreated	Raised when the user draws a new link. (Inherited from DiagramBase.)
	LinkCreating	A validation event raised while the user is drawing a new link. (Inherited from DiagramBase.)
	LinkDeactivated	Raised when a link is no longer the **ActiveItem**. (Inherited from DiagramBase.)
	LinkDeleted	Raised when a link is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDeleting	Raised when the user tries to delete a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkDeselected	Raised when a link is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDoubleClicked	Raised when the user double-clicks a link. (Inherited from DiagramBase.)
	LinkHovered	Raised when the mouse pointer hovers over a link. (Inherited from DiagramBase.)
	LinkModified	Raised when the user moves any control point of a link. (Inherited from DiagramBase.)
	LinkModifyCancelled	Raised when the user cancels move or resize operation being done on a link. (Inherited from DiagramBase.)
	LinkModifying	A validation event raised while the user is modifying a link. (Inherited from DiagramBase.)
	LinkPasted	Raised when a link has been pasted from the clipboard. (Inherited from DiagramBase.)
	LinkRouted	Raised when a link is routed by the auto-routing algorithm. (Inherited from DiagramBase.)
	LinkSelected	Raised when a link is selected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkSelecting	Raised when the user tries to select a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkSplit	Raised when the user splits a link interactively. (Inherited from DiagramBase.)
	LinkStartModifying	A validation event raised when the user starts modifying a link. (Inherited from DiagramBase.)
	LinkTextEdited	Raised when the user edits the text of a link. (Inherited from DiagramBase.)
	LinkTextEditing	A validation event that lets you approve or reject changes made to the text of a link. (Inherited from DiagramBase.)
	MeasureUnitChanged	Raised when the MeasureUnit property value has changed. (Inherited from DiagramBase.)
	NodeActivated	Raised when a node becomes the **ActiveItem**. (Inherited from DiagramBase.)
	NodeClicked	Raised when the user clicks a node. (Inherited from DiagramBase.)
	NodeCreateCancelled	Raised when the user cancels node creation. (Inherited from DiagramBase.)
	NodeCreated	Raised when the user draws a new node. (Inherited from DiagramBase.)
	NodeCreating	A validation event raised while the user is drawing a new node. (Inherited from DiagramBase.)
	NodeDeactivated	Raised when a node is no longer the **ActiveItem**. (Inherited from DiagramBase.)

	NodeDeleted	Raised when a node is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDeleting	Raised when the user tries to delete a node, this event lets you cancel the operation. (Inherited from DiagramBase.)
	NodeDeselected	Raised when a node is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDoubleClicked	Raised when the user double-clicks a node. (Inherited from DiagramBase.)
	NodeHovered	Raised when the mouse pointer hovers over a node. (Inherited from DiagramBase.)
	NodeModified	Raised when the user moves or resizes a node. (Inherited from DiagramBase.)
	NodeModifyCancelled	Raised when the user cancels move or resize operation being done on a node. (Inherited from DiagramBase.)
	NodeModifying	A validation event raised while the user is moving or resizing a node. (Inherited from DiagramBase.)
	NodePasted	Raised when a node has been pasted from the clipboard. (Inherited from DiagramBase.)
	NodeSelected	Raised when a node is selected, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeSelecting	Raised when the user tries to select a node, this event lets you cancel the operation. (Inherited from DiagramBase.)
	NodeStartModifying	A validation event raised when the user starts modifying a node. (Inherited from DiagramBase.)
	NodeTextEdited	Raised when the user edits the text of a node. (Inherited from DiagramBase.)
	NodeTextEditing	A validation event that lets you approve or reject changes made to the text of a node. (Inherited from DiagramBase.)
	OriginAnchorChanged	Raised when the OriginAnchor property value of a link has changed. (Inherited from DiagramBase.)
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation. (Inherited from DiagramBase.)
	QueryAlignTarget	Raised to validate a node as target for alignment guides when AutoAlignNodes is enabled. (Inherited from DiagramBase.)
	Repaint	Raised when the diagram views should be repainted. (Inherited from DiagramBase.)
	RequestCacheReset	Raised when an object within the diagram requests a reset of the cached bitmap. (Inherited from DiagramBase.)
	RequestClipboard	Raised when an object within the diagram requests an operation with the clipboard. (Inherited from DiagramBase.)
	RequestDrawInteractions	For internal use. (Inherited from DiagramBase.)
	RequestTimer	Raised when an object within the diagram needs to start or stop a timer. (Inherited from DiagramBase.)
	SelectionChanged	Raised when an item is added to or removed from the selection (Inherited from DiagramBase.)
	SelectionModified	Raised when the user moves multiple selected items. (Inherited from DiagramBase.)
	SelectionModifying	Raised while the user is moving multiple selected items, this event lets you cancel the modification. (Inherited from DiagramBase.)

	SerializeTag	Raised when the **Tag** of a diagram item must be serialized into an XML document. (Inherited from DiagramBase.)
	SetSelfLoopShape	Raised when a link becomes a self-loop, giving you a chance to override the default look shape. (Inherited from DiagramBase.)
	ShapeModified	Raised when the user modifies the shape of a ShapeNode by moving one of the control points corresponding to shape formula parameters. (Inherited from DiagramBase.)
	ShapeNotDefined	Raised when the diagram being loaded contains nodes with an undefined shape. (Inherited from DiagramBase.)
	TableColumnResized	Raised when the user finishes resizing a column in a table. (Inherited from DiagramBase.)
	TableColumnResizing	Raised when the user resizes a column in a table interactively. (Inherited from DiagramBase.)
	TableRowResized	Raised when the user finishes resizing a row in a table. (Inherited from DiagramBase.)
	TableRowResizing	Raised when the user resizes a row in a table interactively. (Inherited from DiagramBase.)
	TableSectionCollapsed	Raised when the user collapses the rows of a table's section by clicking the [-] button in the section's header row. (Inherited from DiagramBase.)
	TableSectionExpanded	Raised when the user expands the rows of a table's section by clicking the [+] button in the section's header row. (Inherited from DiagramBase.)
	TreeCollapsed	Raised when a user collapses a tree branch by clicking the [-] button of an **Expandable** node. (Inherited from DiagramBase.)
	TreeExpanded	Raised when a user expands a tree branch by clicking the [+] button of an **Expandable** node. (Inherited from DiagramBase.)
	TreeItemTextEditing	A validation event that lets you approve or reject changes made to the text of a TreeViewNode's item. (Inherited from DiagramBase.)
	ValidateAnchorPoint	A validation event that lets you approve or reject connecting a link to an anchor point. (Inherited from DiagramBase.)
	ValidateLinkCrossing	A validation event that lets you approve or reject displaying a decoration at the intersection point of links. (Inherited from DiagramBase.)
	ViewRemoved	Raised when a DiagramView is detached from this diagram. (Inherited from DiagramBase.)

See Also

DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument Constructor

Initializes a new instance of the DiagramDocument class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramDocument ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument Methods

DiagramDocument.Clone Method

Creates a copy of this DiagramDocument.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual DiagramDocument Clone ()
```

Visual Basic Copy Code

```
Public Overridable Function Clone () As DiagramDocument
```

Return Value

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.CopyTo Method

Copies this document's pages to specified target document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void CopyTo (
    DiagramDocument target
)
```

Visual Basic Copy Code

```
Public Overridable Sub CopyTo( _
    target As DiagramDocument _
)
```

Parameters

target
The target document.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromJson Method

Loads the diagram document from a JSON-formatted string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void LoadFromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Overridable Sub LoadFromJson( _
    json As String _
)
```

Parameters

json

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromJsonAsync Method

Loads the diagram document from a JSON-formatted string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task LoadFromJsonAsync (
    string json
)
```

Visual Basic Copy Code

```
Public Function LoadFromJsonAsync( _
    json As String _
) As Task
```

Parameters

json

Return Value

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromJsonFile Method

Loads the diagram document from a JSON file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromJsonFile (
    string fileName
)
```

Visual Basic Copy Code

```
Public Sub LoadFromJsonFile( _
    fileName As String _
)
```

Parameters

fileName

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromJsonFileAsync Method

Loads the diagram document from a JSON file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task LoadFromJsonFileAsync (
    string fileName
)
```

Visual Basic Copy Code

```
Public Function LoadFromJsonFileAsync( _
    fileName As String _
) As Task
```

Parameters

fileName

Return Value

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromJsonObject Method

Loads the diagram document from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void LoadFromJsonObject (
    JsonObject documentObj
)
```

Visual Basic Copy Code

```
Public Overridable Sub LoadFromJsonObject( _
    documentObj As JsonObject _
)
```

Parameters

documentObj
A JsonObject representing the document.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromString Method

Loads the document from the specified string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromString (
    string str,
    SaveToStringFormat format
)
```

Visual Basic Copy Code

```
Public Sub LoadFromString( _
    str As String, _
    format As SaveToStringFormat _
)
```

Parameters

str
The string to load from.

format

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromXml Method

Loads the document from an XML document or file.

Overload List

Name	Description
DiagramDocument.LoadFromXml (String)	Loads the document from the specified XML file.
DiagramDocument.LoadFromXml (String, Boolean)	Loads the document from the specified XML file.
DiagramDocument.LoadFromXml (XmlDocument)	Loads the document from the specified **XmlDocument** object.
DiagramDocument.LoadFromXml (XmlDocument, Boolean)	Loads the document from the specified **XmlDocument** object.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromXml Method (String)

Loads the document from the specified XML file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    string fileName
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    fileName As String _
)
```

Parameters

fileName
The name of the XML file the data should be read from.

See Also

LoadFromXml Method Overload List
DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromXml Method (String, Boolean)

Loads the document from the specified XML file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    string fileName,
    bool loadDiagramProperties
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    fileName As String, _
    loadDiagramProperties As Boolean _
)
```

Parameters

fileName
The name of the XML file the data should be read from.

loadDiagramProperties
true to load values of diagram properties; **false** to load only diagram items.

See Also

LoadFromXml Method Overload List
DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromXml Method (XmlDocument)

Loads the document from the specified **XmlDocument** object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    XmlDocument document
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    document As XmlDocument _
)
```

Parameters

document
A .NET **XmlDocument** object from which to load the diagram data.

See Also

LoadFromXml Method Overload List
DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromXml Method (XmlDocument, Boolean)

Loads the document from the specified **XmlDocument** object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    XmlDocument document,
    bool loadDiagramProperties
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    document As XmlDocument, _
    loadDiagramProperties As Boolean _
)
```

Parameters

document
A .NET **XmlDocument** object from which to load the diagram data.

loadDiagramProperties
true to load values of diagram properties; **false** to load only diagram items.

See Also

LoadFromXml Method Overload List
DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromXmlAsync Method

Overload List

Name	Description
DiagramDocument.LoadFromXmlAsync (String)	Loads the document from the specified XML file.
DiagramDocument.LoadFromXmlAsync (XmlDocument)	Loads the diagram document from an XML document.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromXmlAsync Method (String)

Loads the document from the specified XML file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task LoadFromXmlAsync (
    string fileName
)
```

Visual Basic Copy Code

```
Public Function LoadFromXmlAsync( _
    fileName As String _
) As Task
```

Parameters

fileName
The name of the XML file the data should be read from.

Return Value

See Also

LoadFromXmlAsync Method Overload List
DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.LoadFromXmlAsync Method (XmlDocument)

Loads the diagram document from an XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task LoadFromXmlAsync (
    XmlDocument document
)
```

Visual Basic Copy Code

```
Public Function LoadFromXmlAsync( _
    document As XmlDocument _
) As Task
```

Parameters

document

Return Value

See Also

LoadFromXmlAsync Method Overload List
DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToJson Method

Saves the diagram document to a JSON-formatted string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual string SaveToJson ()
```

Visual Basic Copy Code

```
Public Overridable Function SaveToJson () As String
```

Return Value

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToJsonAsync Method

Saves the diagram document to a JSON-formatted string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task<string> SaveToJsonAsync ()
```

Visual Basic Copy Code

```
Public Function SaveToJsonAsync () As Task(Of String)
```

Return Value

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToJsonFile Method

Saves the diagram document into a JSON file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToJsonFile (
    string fileName
)
```

Visual Basic Copy Code

```
Public Sub SaveToJsonFile( _
    fileName As String _
)
```

Parameters

fileName

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToJsonFileAsync Method

Saves the diagram document into a JSON file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task SaveToJsonFileAsync (
    string fileName
)
```

Visual Basic Copy Code

```
Public Function SaveToJsonFileAsync( _
    fileName As String _
) As Task
```

Parameters

fileName

Return Value

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToJsonObject Method

Saves the diagram document to a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void SaveToJsonObject (
    JsonObject documentObj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Public Overridable Sub SaveToJsonObject( _
    documentObj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

documentObj
A JsonObject representing the document.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToString Method

Saves the document as an ASCII string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string SaveToString (
    SaveToStringFormat format,
    bool clearDirty
)
```

Visual Basic Copy Code

```
Public Function SaveToString( _
    format As SaveToStringFormat, _
    clearDirty As Boolean _
) As String
```

Parameters

format
One of the SaveToStringFormat values.

clearDirty
Indicates whether the 'dirty' flag should be cleared.

Return Value

A string value.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToXml Method

Saves the document to an XML document or file.

Overload List

Name	Description
DiagramDocument.SaveToXml (String)	Saves the document to the specified file in an XML format.
DiagramDocument.SaveToXml (XmlDocument)	Saves the document to the specified **XmlDocument** object.
DiagramDocument.SaveToXml (XmlDocument, Boolean)	Saves the document to the specified **XmlDocument** object.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToXml Method (String)

Saves the document to the specified file in an XML format.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToXml (
    string fileName
)
```

Visual Basic Copy Code

```
Public Sub SaveToXml( _
    fileName As String _
)
```

Parameters

fileName
The full path and name of the disk file where the document should be saved.

See Also

SaveToXml Method Overload List
DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToXml Method (XmlDocument)

Saves the document to the specified **XmlDocument** object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToXml (
    XmlDocument document
)
```

Visual Basic Copy Code

```
Public Sub SaveToXml( _
    document As XmlDocument _
)
```

Parameters

document
A .NET **XmlDocument** object where the document should be saved.

See Also

SaveToXml Method Overload List
DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToXml Method (XmlDocument, Boolean)

Saves the document to the specified **XmlDocument** object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToXml (
    XmlDocument document,
    bool includeUnalteredProperties
)
```

Visual Basic Copy Code

```
Public Sub SaveToXml( _
    document As XmlDocument, _
    includeUnalteredProperties As Boolean _
)
```

Parameters

document
A .NET **XmlDocument** object where the document should be saved.

includeUnalteredProperties
true if properties that have their default value should be saved; otherwise, **false**.

See Also

SaveToXml Method Overload List
DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToXmlAsync Method

Overload List

Name	Description
DiagramDocument.SaveToXmlAsync (String)	Saves the diagram document into an XML file.
DiagramDocument.SaveToXmlAsync (XmlDocument)	Saves the diagram document into an XML document.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToXmlAsync Method (String)

Saves the diagram document into an XML file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task SaveToXmlAsync (
    string fileName
)
```

Visual Basic Copy Code

```
Public Function SaveToXmlAsync( _
    fileName As String _
) As Task
```

Parameters

fileName

Return Value

See Also

SaveToXmlAsync Method Overload List
DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.SaveToXmlAsync Method (XmlDocument)

Saves the diagram document into an XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Task SaveToXmlAsync (
    XmlDocument document
)
```

Visual Basic Copy Code

```
Public Function SaveToXmlAsync( _
    document As XmlDocument _
) As Task
```

Parameters

document

Return Value

See Also

SaveToXmlAsync Method Overload List
DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument Properties

DiagramDocument.Dirty Property

Gets a value indicating whether the document has changed since the last save.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Dirty { get; }
```

Visual Basic

```
Public ReadOnly Property Dirty As Boolean
```

Property Value

true if the document has changed; otherwise, **false**.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument.Pages Property

Gets a list of the diagram pages contained in this document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public ObservableCollection<DiagramPage> Pages { get; }
```

Visual Basic

```
Public ReadOnly Property Pages As ObservableCollection(Of DiagramPage)
```

Property Value

A list with all document pages.

See Also

DiagramDocument Members
DiagramDocument Class
MindFusion.Diagramming Namespace

DiagramDocument Events

DiagramEventArgs Class

Contains the arguments passed to event handlers, such as Clicked and DoubleClicked.

Syntax

C#

```
public class DiagramEventArgs : EventArgs
```

Visual Basic

```
Public Class DiagramEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.DiagramEventArgs

See Also

DiagramEventArgs Members
MindFusion.Diagramming Namespace

DiagramEventArgs Members

Contains the arguments passed to event handlers, such as Clicked and DoubleClicked.

The following tables list the members exposed by the DiagramEventArgs type.

Public Constructors

	Name	Description
	DiagramEventArgs	Overloaded.

Public Properties

	Name	Description
	ClipRect	Gets the current clipping rectangle.
	Graphics	Gets an object implementing the IGraphics interface whose methods should be used to draw the custom graphics.
	MouseButton	Gets or sets which mouse button has been pressed.
	MousePosition	Gets or sets the current mouse pointer position.
	VisibleRect	Specifies what part of the diagram is currently visible.

See Also

DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs Constructor

Overload List

Name	Description
DiagramEventArgs (IGraphics, Rect, Rect)	Initializes a new instance of the DiagramEventArgs class.
DiagramEventArgs (MouseButton, Point)	Initializes a new instance of the **DiagramEventArgs** class.

See Also

DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs Constructor (IGraphics, Rect, Rect)

Initializes a new instance of the DiagramEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramEventArgs (
    IGraphics graphics,
    Rect visibleRect,
    Rect clipRect
)
```

Visual Basic Copy Code

```
Public New ( _
    graphics As IGraphics, _
    visibleRect As Rect, _
    clipRect As Rect _
)
```

Parameters

graphics

visibleRect

clipRect
A **Microsoft.Maui.Graphics.Rect** specifying the current clipping rectangle.

See Also

DiagramEventArgs Constructor Overload List
DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs Constructor (MouseButton, Point)

Initializes a new instance of the DiagramEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramEventArgs (
    MouseButton mouseButton,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public New ( _
    mouseButton As MouseButton, _
    mousePosition As Point _
)
```

Parameters

mouseButton
The mouse button related to the event.

mousePosition
The cursor location at the time the event is raised, expressed in document coordinates.

See Also

DiagramEventArgs Constructor Overload List
DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs Properties

DiagramEventArgs.ClipRect Property

Gets the current clipping rectangle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect ClipRect { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ClipRect As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** instance.

Remarks

This property is used as an argument to DrawBackground event handlers.

See Also

DiagramEventArgs Members
DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs.Graphics Property

Gets an object implementing the IGraphics interface whose methods should be used to draw the custom graphics.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IGraphics Graphics { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Graphics As IGraphics
```

Property Value

An instance of a class implementing the IGraphics interface.

Remarks

This property is used as an argument to DrawBackground event handlers.

See Also

DiagramEventArgs Members
DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs.MouseButton Property

Gets or sets which mouse button has been pressed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MouseButton MouseButton { get; set; }
```

Visual Basic Copy Code

```
Public Property MouseButton As MouseButton
```

Property Value

A member of the MouseButton enumeration specifying the currently pressed mouse button.

See Also

DiagramEventArgs Members
DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs.MousePosition Property

Gets or sets the current mouse pointer position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point MousePosition { get; set; }
```

Visual Basic Copy Code

```
Public Property MousePosition As Point
```

Property Value

A **Point** value specifying the current mouse position.

See Also

DiagramEventArgs Members
DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramEventArgs.VisibleRect Property

Specifies what part of the diagram is currently visible.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect VisibleRect { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property VisibleRect As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** instance.

Remarks

This property is used as an argument to DrawBackground event handlers.

See Also

DiagramEventArgs Members
DiagramEventArgs Class
MindFusion.Diagramming Namespace

DiagramItem Class

Represents an item in the diagram document. All classes representing diagram elements derive from **DiagramItem**.

Syntax

C# Copy Code

```
public abstract class DiagramItem : IDisposable, InplaceEditable
```

Visual Basic Copy Code

```
Public MustInherit Class DiagramItem
    Implements IDisposable, InplaceEditable
```

Remarks

All items in the diagram document - links and nodes - share common base class called **DiagramItem**. Its main function is to provide interface for the most common methods and properties of every diagram element.

MvcDiagram invokes the Draw method when an item must be drawn on screen. You can call **Draw** if you need to draw an item on your own graphics surface. Override the method when implementing custom item types to provide your own rendering functionality. The standard item types draw their interior and borders using the objects assigned to the Brush and Pen properties. The font of the item's text is set by the Font property.

The order in which items are painted from bottom to top is called the z-order. Moving items up or down the z-order can be done by means of the ZBottom, ZTop, ZLevelUp and ZLevelDown methods. Accessing and setting the index of an item in the z-order can be done using the ZIndex property.

A hierarchy of items can be built, so that moving an item that is higher in the hierarchy moves all subordinated items too. The immediate subordinates of an item are represented by a Group instance. The group of items attached to another item is accessed through the SubordinateGroup property of the latter. MasterGroup gets the group in which an item is subordinate.

Every item can cast a shadow, whose offset from the item is specified by the ShadowOffsetX and ShadowOffsetY properties. The shadow is painted using the color set in ShadowBrush.

An application can associate custom data with an item by means of the Tag property. The tags are only meant to hold data related to the entities represented by the chart items. They are not used or modified by MvcDiagram in any way. The tags are saved and loaded together with the diagram elements when serializing the document to a disk file or a stream.

Items can be locked for interactive modification through the Locked property. Locked items can still be modified programmatically. Items can also be marked as locked for automatic layout methods through the IgnoreLayout property.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramLink
    MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.Selection
    MindFusion.Diagramming.DiagramLink
    MindFusion.Diagramming.DiagramNode
    MindFusion.Diagramming.Selection

See Also

DiagramItem Members

Represents an item in the diagram document. All classes representing diagram elements derive from DiagramItem.

The following tables list the members exposed by the **DiagramItem** type.

Protected Constructors

	Name	Description
	DiagramItem	Overloaded. Initializes a new instance of the **DiagramItem** class.

Protected Fields

	Name	Description
	disposed	For internal use.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of this item.
	CssClass	Gets or sets the CSS class associated with this item.
	EffectiveBrush	Gets the effective brush of the item.
	EffectiveFont	Gets the currently effective font of the item.
	EffectivePen	Gets the effective pen of the item.
	EffectiveShadowBrush	Gets the effective shadow brush of the item.
	EffectiveTextBrush	Gets the effective text brush of the item.
	Font	Gets or sets the font used to render the item's text.
	HyperLink	Gets or sets the hyperlink associated with this diagram item.
	Id	Gets or sets user-defined identifier associated with this item.
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods.
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer.
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**.
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item.
	Opacity	Gets or sets the opacity of this item.
	Parent	Gets the diagram that contains this item.
	Pen	Gets or sets the pen used to paint the frame of the item.
	Printable	Gets or sets a value indicating whether to print the item.
	RuntimeIndex	For internal use.
	Selected	Gets or sets a value indicating whether a diagram item is selected.

	ShadowBrush	Gets or sets the brush used to paint the item's shadow.
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item.
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item.
	Style	Gets or sets the style associated with the item.
	SubordinateGroup	Gets a reference to the group of items that are attached to this one.
	Tag	Gets or sets user-defined data associated with this item.
	Text	Gets or sets the text displayed by this item.
	TextBrush	Gets or sets the brush used to paint texts inside the item.
	TextFormat	Gets or sets the alignment of text displayed inside this item.
	TextPadding	Gets or sets the spacing between the item boundaries and its text.
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element.
	TopLevel	True if this is a top-level item in the diagram, or false if it is inside a container.
	Visible	Gets or sets a value indicating whether this item is visible.
	Weight	Gets or sets a weight value used in layout and path-finding algorithms.
	ZIndex	Gets or sets the z-order position of this item.

Protected Properties

	Name	Description
	MeasureUnit	Returns the diagram measure unit.

Public Methods

	Name	Description
	Animate	Animates the item.
	Clone	Creates a copy of this item.
	ContainsPoint	Overloaded. Determines whether this item contains the specified point.
	Dispose	Overloaded.
	DoModify	For internal use.
	Draw	Draws the item on the specified IGraphics surface.
	DrawHandles	Renders the adjustment handles of this item.
	DrawInRect	
	DrawShadow	Draws the shadow of this item on the specified IGraphics surface.
	GetBounds	Gets the bounding rectangle of this item.
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item.
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item.
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item.

	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item.
	GetOutlinePoly	Gets a polygon approximating the outline of this object.
	GetRepaintRect	Gets the repaint region for this item, taking into account factors such as pen, selection handles and shadow.
	GetRotatedBounds	Gets the bounding rectangle of this item when it is rotated.
	GetToolTip	Gets the tooltip that should be displayed for this item at the specified point.
	HitTest	Returns the diagram item that contains the specified point.
	HitTestHandle	Determines whether a selection handle of this item is located at the specified point.
	HitTestManipulators	Finds the **Manipulator** object located at the specified point.
	LabelFromPoint	Returns the ItemLabel located at specified point.
	NotInteractive	Gets whether the user is prevented from manipulating this item.
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user.
	OnLoad	Called after successful deserialization.
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression.
	RestoreItemProperties	For internal use.
	SaveItemProperties	For internal use.
	ZBottom	Moves the item to the bottom of the z-order.
	ZLevelDown	Moves the item one level down in the z-order.
	ZLevelUp	Moves the item one level up in the z-order.
	ZTop	Moves the item to the top of the z-order.

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object.
	AllowCreate	Invoked during interactive creation of items to test whether the operation can be completed in the current context.
	AllowModify	Invoked during interactive modification of items to test whether the operation can be completed in the current context.
	CancelModify	Invoked when the interactive modification of this object has been canceled.
	CompleteCreate	Invoked when the item creation has been completed.
	CompleteModify	Invoked when the item modification has been completed.
	CreateProperties	Creates an item property set corresponding to the type of this diagram item.
	CreateState	Creates an item state corresponding to the type of this diagram item.
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics.
	Finalize	DiagramItem finalizer.

	FreeResources	Called when the item is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history.
	GetValue	Gets the effective value of the style property with the specified name.
	LoadFromJson	Loads the item from a JSON object.
	LoadFromXml	Loads the item content from an XML element.
	LoadObsolete	For internal use.
	OnAdd	Called when the item is added to a Diagram, either by the user, programmatically or via undo / redo.
	OnChildModified	Called when a group child node has been modified.
	OnClick	Invoked when the user clicks this item.
	OnDoubleClick	Invoked when the user double-clicks this item.
	OnMeasureUnitChanged	Called when the diagram's MeasureUnit changes.
	OnRemove	Called when an item is removed from a **Diagram**, either by the user, programmatically or via undo / redo.
	OnStyleChanged	Invoked when the associated style or theme have changed.
	OnTextChanged	Called when the value of the Text property changes.
	OnTextFormatChanged	Called when the value of the TextFormat property changes.
	OnZoomChanged	Called when the diagram's **ZoomFactor** changes.
	RemoveManipulator	Removes the specified manipulator from the object.
	Repaint	Overloaded.
	RestoreProperties	Restores the properties of this diagram item.
	RestoreState	Restores the state of this diagram item.
	SaveProperties	Saves the properties of this diagram item.
	SaveState	Saves the state of this diagram item.
	SaveToJson	Saves the item as a JSON object.
	SaveToXml	Saves the item content into an XML element.
	SetDiagramDirty	Marks the diagram as changed.
	SetExpanded	Sets whether the item is in expanded or collapsed state.
	ShouldRender	Determines whether the item should be rendered in the current context.
	StartCreate	Invoked once when the user starts creating a new item interactively.
	StartModify	Invoked when the user starts modifying this item interactively.
	UpdateCreate	Invoked during interactive creation of this item.
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster.
	UpdateModify	Invoked during interactive modification of this item.

See Also

DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem Fields

DiagramItem.disposed Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected bool disposed
```

Visual Basic Copy Code

```
Protected disposed As Boolean
```

Remarks

This field is used in the implementation of the IDisposable pattern.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem Constructor

Initializes a new instance of the DiagramItem class.

Overload List

Name	Description
DiagramItem ()	Initializes a new instance of the **DiagramItem** class.
DiagramItem (Diagram)	Initializes a new instance of the **DiagramItem** class with default values supplied form the specified Diagram.
DiagramItem (DiagramItem)	Initializes a new instance of the **DiagramItem** class using the specified diagram item as a prototype.

See Also

DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem Constructor ()

Initializes a new instance of the DiagramItem class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected DiagramItem ()
```

Visual Basic

```
Protected Sub New ()
```

Remarks

This constructor sets the DiagramItem properties to their default values.

See Also

DiagramItem Constructor Overload List
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem Constructor (Diagram)

Initializes a new instance of the DiagramItem class with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected DiagramItem (
    Diagram parent
)
```

Visual Basic

```
Protected New ( _
    parent As Diagram _
)
```

Parameters

parent
The Diagram from which to obtain default values for the item properties.

Remarks

This constructor initializes the DiagramItem properties with the values specified in the respective default-value properties of the Diagram object. For more information see **Initial Values of Item Properties**.

See Also

DiagramItem Constructor Overload List
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem Constructor (DiagramItem)

Initializes a new instance of the DiagramItem class using the specified diagram item as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected DiagramItem (
    DiagramItem prototype
)
```

Visual Basic

```
Protected New ( _
    prototype As DiagramItem _
)
```

Parameters

prototype
The item to use as a prototype. The properties of the new item are initialized from the property values of this item.

Remarks

This constructor copies the DiagramItem property values from the specified prototype item.

See Also

DiagramItem Constructor Overload List
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem Methods

DiagramItem.AddManipulator Method

Adds the specified manipulator to this object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void AddManipulator (
    Manipulator mnp
)
```

Visual Basic Copy Code

```
Protected Sub AddManipulator( _
    mnp As Manipulator _
)
```

Parameters

mnp
An instance of **Manipulator** -derived class.

Remarks

Use this method to add custom manipulators to diagram elements. Manipulators are interactive widgets associated with an item that let users change aspects of the item different from its position and size. Some built-in manipulators are the +/- icons displayed for Expandable nodes and the scrollbar of a TreeViewNode.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
RemoveManipulator Method

DiagramItem.AllowCreate Method

Invoked during interactive creation of items to test whether the operation can be completed in the current context.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected abstract bool AllowCreate (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected MustOverride Function AllowCreate( _
    current As Point, _
    ist As InteractionState _
) As Boolean
```

Parameters

current

The current mouse pointer position.

ist

An InteractionState object containing more information about the current state of the operation.

Return Value

true to allow the creation of this item; otherwise, **false**.

Remarks

This method is invoked after each mouse movement in order to update the mouse pointer icon, so do not perform any time-consuming operations in its implementation.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.AllowModify Method

Invoked during interactive modification of items to test whether the operation can be completed in the current context.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected abstract bool AllowModify (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected MustOverride Function AllowModify( _
    current As Point, _
    ist As InteractionState _
) As Boolean
```

Parameters

current

The current mouse pointer position.

ist

An InteractionState object containing more information about the current state of the operation.

Return Value

true to allow the modification of this item; otherwise, **false**.

Remarks

This method is invoked after each mouse movement in order to update the mouse pointer icon, so do not perform any time-consuming operations in its implementation.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Animate Method

Animates the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Animate (
    ItemAnimation animation
)
```

Visual Basic Copy Code

```
Public Sub Animate( _
    animation As ItemAnimation _
)
```

Parameters

animation
An ItemAnimation -derived object.

Remarks

Item's animated values are considered transient and do not change actual property values. If you want to assign last value from animation to actual property such as Bounds or Opacity, set the KeepLastValue flag in AnimationOptions as in example below.

Example

C# Copy Code

```
node.Animate(
    new PathAnimation(
        new List<Point>
        {
            new Point(30, 30),
            new Point(50, 30),
            new Point(75, 66),
            new Point(125, 88)
        },
        new AnimationOptions
        {
            KeepLastValue = true
        }));
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Animations

DiagramItem.CancelModify Method

Invoked when the interactive modification of this object has been canceled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void CancelModify (
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overridable Sub CancelModify( _
    ist As InteractionState _
)
```

Parameters

ist
An InteractionState object containing more information about the current state of the operation.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Clone Method

Creates a copy of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual DiagramItem Clone (
    bool clipboard
)
```

Visual Basic Copy Code

```
Public Overridable Function Clone( _
    clipboard As Boolean _
) As DiagramItem
```

Parameters

clipboard
true if the copy will be stored in the Windows clipboard; otherwise, **false**.

Return Value

The copy of this item.

Remarks

The default implementation of this item invokes the item's class copy constructor through reflection. If your custom type does not provide a copy constructor, you must implement this method to provide for clipboard operations support.

See Also

DiagramItem Members
DiagramItem Class

DiagramItem.CompleteCreate Method

Invoked when the item creation has been completed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void CompleteCreate (
    Point end
)
```

Visual Basic Copy Code

```
Protected Overridable Sub CompleteCreate( _
    end As Point _
)
```

Parameters

end
The mouse pointer position.

Remarks

This method is invoked when the user releases the mouse button while drawing a new item.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.CompleteModify Method

Invoked when the item modification has been completed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void CompleteModify (
    Point end,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overridable Sub CompleteModify( _
    end As Point, _
    ist As InteractionState _
)
```

Parameters

end
The mouse pointer position.

ist
An InteractionState object containing more information about the current state of the operation.

Remarks

This method is invoked when the user releases the mouse button while modifying an item.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.ContainsPoint Method

Determines whether this item contains the specified point.

Overload List

Name	Description
DiagramItem.ContainsPoint (Point)	Determines whether this item contains the specified point.
DiagramItem.ContainsPoint (Point, Double)	Determines whether this item contains the specified point.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.ContainsPoint Method (Point)

Determines whether this item contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract bool ContainsPoint (
    Point point
)
```

Visual Basic Copy Code

```
Public MustOverride Function ContainsPoint( _
    point As Point _
) As Boolean
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying the logical coordinates of a diagram point.

Return Value

true it this item contains the specified point; otherwise, **false**.

Remarks

This method is used during hit-testing to see if the mouse is inside the item.

See Also

ContainsPoint Method Overload List
DiagramItem Members
DiagramItem Class

DiagramItem.ContainsPoint Method (Point, Double)

Determines whether this item contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract bool ContainsPoint (
    Point point,
    double threshold
)
```

Visual Basic Copy Code

```
Public MustOverride Function ContainsPoint( _
    point As Point, _
    threshold As Double _
) As Boolean
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying the logical coordinates of a diagram point.

threshold
Specifies the size of the area around the item outlines where the point is still considered inside the item.

Return Value

true it this item contains the specified point; otherwise, **false**.

See Also

ContainsPoint Method Overload List
DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
HitTest Method

DiagramItem.CreateProperties Method

Creates an item property set corresponding to the type of this diagram item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual DiagramItemProperties CreateProperties ()
```

Visual Basic Copy Code

```
Protected Overridable Function CreateProperties () As DiagramItemProperties
```

Return Value

A DiagramItemProperties -derived object containing a snapshot of the item's properties.

Remarks

This method is invoked by the undo/redo subsystem when creating a ChangeItemCmd instance.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.CreateState Method

Creates an item state corresponding to the type of this diagram item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual DiagramItemState CreateState ()
```

Visual Basic Copy Code

```
Protected Overridable Function CreateState () As DiagramItemState
```

Return Value

An instance of a class derived from DiagramItemState.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
SaveState Method
RestoreState Method

DiagramItem.Dispose Method

Overload List

Name	Description
DiagramItem.Dispose ()	Disposes the resources managed by this item.
DiagramItem.Dispose (Boolean)	Disposes any disposable resources managed by this item.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Dispose Method ()

Disposes the resources managed by this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Dispose ()
```

Visual Basic Copy Code

```
Public Sub Dispose ()
```

Remarks

This method is used in the implementation of the IDisposable pattern.

See Also

Dispose Method Overload List
DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Dispose Method (Boolean)

Disposes any disposable resources managed by this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void Dispose (
    bool disposing
)
```

Visual Basic Copy Code

```
Protected Overridable Sub Dispose( _
    disposing As Boolean _
)
```

Parameters

disposing
Specifies whether the method is called explicitly or from the garbage collector.

Remarks

This method is used in the implementation of the IDisposable pattern.

See Also

Dispose Method Overload List
DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.DoModify Method

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool DoModify (
    Point start,
```

```
    Point end,
    AdjustmentHandle handle
)
```

Visual Basic Copy Code

```
Public Function DoModify( _
    start As Point, _
    end As Point, _
    handle As AdjustmentHandle _
) As Boolean
```

Parameters

start

end

handle

Return Value

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Draw Method

Draws the item on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract void Draw (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public MustOverride Sub Draw( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
The IGraphics surface to draw the item on.

options
A RenderOptions object specifying options for the drawing operation.

Remarks

MvcDiagram calls this method to draw a representation of the diagram item onto the specified graphics surface. Implement this method in custom node or link types if you need to provide your own drawing functionality. The method is called when rendering the diagram on screen, or exporting it to PDF or SVG files. The IGraphics methods call the respective Microsoft.Maui.Graphics API methods when drawing on screen, or generate PDF of SVG tags when using PdfExporter and SvgExporter to export diagram files.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.DrawHandles Method

Renders the adjustment handles of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract void DrawHandles (
    IGraphics graphics,
    HandlesVisualStyle style,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public MustOverride Sub DrawHandles( _
    graphics As IGraphics, _
    style As HandlesVisualStyle, _
    options As RenderOptions _
)
```

Parameters

graphics
The IGraphics surface on which the handles should be drawn.

style
A HandlesVisualStyle instance containing brush and pen objects that should be used when drawing handles.

options
Provides options for the drawing operation.

Remarks

This method is invoked when the item's adjustment handles must be rendered.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
HitTestHandle Method

DiagramItem.DrawInRect Method

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void DrawInRect (
    IGraphics graphics,
    RenderOptions options,
    Rect target
)
```

Visual Basic Copy Code

```
Public Overridable Sub DrawInRect( _
    graphics As IGraphics, _
    options As RenderOptions, _
    target As Rect _
)
```

Parameters

graphics

options

target

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.DrawManipulators Method

Draws the associated manipulators on a specified graphics.

Overload List

Name	Description
DiagramItem.DrawManipulators (IGraphics, Boolean)	Draws the associated manipulators on the specified graphics.
DiagramItem.DrawManipulators (IGraphics, Boolean, Boolean)	Draws the associated manipulators in the specified graphics.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.DrawManipulators Method (IGraphics, Boolean)

Draws the associated manipulators on the specified graphics.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void DrawManipulators (
    IGraphics graphics,
    bool clip
)
```

Visual Basic Copy Code

```
Protected Sub DrawManipulators( _
    graphics As IGraphics, _
    clip As Boolean _
)
```

Parameters

graphics
The IGraphics surface to draw the manipulators on.

clip
true to render only inside the item's bounds; otherwise, **false**.

Remarks

This method is invoked to render the item's manipulators, for example, collapse/expand buttons, scrollbars, and so on.

See Also

DrawManipulators Method Overload List
DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.DrawManipulators Method (IGraphics, Boolean, Boolean)

Draws the associated manipulators in the specified graphics.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void DrawManipulators (
    IGraphics graphics,
    bool clip,
    bool inLocal
)
```

Visual Basic Copy Code

```
Protected Sub DrawManipulators( _
    graphics As IGraphics, _
    clip As Boolean, _
    inLocal As Boolean _
)
```

Parameters

graphics
The IGraphics surface to draw the manipulators on.

clip
true to render only inside the item's bounds; otherwise, **false**.

inLocal
true if current graphics are in local coordinates; otherwise, **false**.

Remarks

This method is invoked to render the item's manipulators, for example, collapse/expand buttons, scrollbars, and so on.

See Also

DrawManipulators Method Overload List
DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.DrawShadow Method

Draws the shadow of this item on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract void DrawShadow (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public MustOverride Sub DrawShadow( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
The IGraphics surface to draw the shadow on.

options
A RenderOptions object specifying options for the drawing operation.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Finalize Method

DiagramItem finalizer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void Finalize ()
```

Visual Basic Copy Code

```
Protected Overrides Sub Finalize ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.FreeResources Method

Called when the item is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void FreeResources ()
```

Visual Basic Copy Code

```
Protected Overridable Sub FreeResources ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.GetBounds Method

Gets the bounding rectangle of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Rect GetBounds ()
```

Visual Basic Copy Code

```
Public Overridable Function GetBounds () As Rect
```

Return Value

The rectangle that bounds this item.

Remarks

The Bounds property getter returns the result of this method. MvcDiagram treats the returned value as the logical coordinates of the item and uses it in many situations when comparing items geometrically, when looking for items intersections, and so on.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.GetEffectiveBrush Method

Creates a Brush object representing the currently effective brush of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush GetEffectiveBrush ()
```

Visual Basic Copy Code

```
Public Function GetEffectiveBrush () As Brush
```

Return Value

A Brush instance.

Remarks

This method returns either the item's Brush if the item's **Brush** is not **null**, or the Brush value of the item's style.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.GetEffectivePen Method

Returns a Pen object representing the currently effective pen of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen GetEffectivePen (
    IGraphics graphics
)
```

Visual Basic Copy Code

```
Public Function GetEffectivePen( _
    graphics As IGraphics _
) As Pen
```

Parameters

graphics

Return Value

Remarks

This method returns either the item's Pen if the item's Pen is not null, or a pen with the Stroke and StrokeThickness properties of the item's style.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.GetEffectiveShadowBrush Method

Returns a Brush object representing the currently effective shadow brush of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush GetEffectiveShadowBrush ()
```

Visual Basic Copy Code

```
Public Function GetEffectiveShadowBrush () As Brush
```

Return Value

A Brush instance.

Remarks

This method returns either the item's ShadowBrush if the item's **ShadowBrush** is not **null**, or the ShadowBrush property of the item's style.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.GetEffectiveTextBrush Method

Returns a Brush object representing the currently effective text brush of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush GetEffectiveTextBrush ()
```

Visual Basic Copy Code

```
Public Function GetEffectiveTextBrush () As Brush
```

Return Value

A Brush instance.

Remarks

This method returns either the item's TextBrush if the item's **TextBrush** is not **null**, or the TextBrush property of the item's style.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.GetOutlinePoly Method

Gets a polygon approximating the outline of this object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual PointCollection GetOutlinePoly (
    bool rendering
)
```

Visual Basic Copy Code

```
Public Overridable Function GetOutlinePoly( _
    rendering As Boolean _
) As PointCollection
```

Parameters

rendering
Specifies whether the outline will be used for rendering (true) or hit-testing (false).

Return Value

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.GetRepaintRect Method

Gets the repaint region for this item, taking into account factors such as pen, selection handles and shadow.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Rect GetRepaintRect (
    bool includeConnected
)
```

Visual Basic Copy Code

```
Public Overridable Function GetRepaintRect( _
    includeConnected As Boolean _
) As Rect
```

Parameters

includeConnected
true if related items should be considered too; otherwise, **false**.

Return Value

A rectangular region that should be repainted when this item is modified.

Remarks

This method is invoked when an item is modified by the user, or when an appearance property of the item changes. The returned region is invalidated and redrawn at the next paint operation.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.GetRotatedBounds Method

Gets the bounding rectangle of this item when it is rotated.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Rect GetRotatedBounds ()
```

Visual Basic Copy Code

```
Public Overridable Function GetRotatedBounds () As Rect
```

Return Value

A rectangle that bounds the rotated shape of the item.

Remarks

This method is invoked during hit-testing to do a quick test whether a point is within the item's bounds, before calling the possibly more time-consuming ContainsPoint method.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.GetToolTip Method

Gets the tooltip that should be displayed for this item at the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual string GetToolTip (
    Point point
)
```

Visual Basic Copy Code

```
Public Overridable Function GetToolTip( _
    point As Point _
) As String
```

Parameters

point
The mouse pointer position.

Return Value

A string specifying the tooltip text.

Remarks

You can override this method to return different tooltips for different regions of the item.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.GetValue Method

Gets the effective value of the style property with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected Object GetValue (
    string name
)
```

Visual Basic Copy Code

```
Protected Function GetValue( _
    name As String _
) As Object
```

Parameters

name
The name of the property to get.

Return Value

An object representing the effective property value.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.HitTest Method

Returns the diagram item that contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual DiagramItem HitTest (
    Point point,
    double threshold,
    bool onlyNodes,
    bool onlyInteractive,
    bool includeHandles,
    bool includeLabels
)
```

Visual Basic Copy Code

```
Public Overridable Function HitTest( _
    point As Point, _
    threshold As Double, _
    onlyNodes As Boolean, _
    onlyInteractive As Boolean, _
    includeHandles As Boolean, _
    includeLabels As Boolean _
) As DiagramItem
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** value specifying the logical coordinates of a diagram point.

threshold
Specifies the size of the area around the item outlines where the point is still considered inside the item.

onlyNodes
true to exclude child links from the search, or **false** otherwise.

onlyInteractive
true to exclude Locked items from the search, or **false** otherwise.

includeHandles
true to also consider adjustment handle locations (possibly outside of node's bounds), or **false** otherwise.

includeLabels
true to also consider item label locations (possibly outside of item's bounds).

Return Value

The item that contains the specified point, or **null** if none found.

Remarks

This method performs recursive search if items contain child items.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
ContainsPoint Method

DiagramItem.HitTestHandle Method

Determines whether a selection handle of this item is located at the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract AdjustmentHandle HitTestHandle (
    Point point,
    double distance
)
```

Visual Basic Copy Code

```
Public MustOverride Function HitTestHandle( _
    point As Point, _
    distance As Double _
) As AdjustmentHandle
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** value specifying where to look for a selection handle.

distance
Maximum distance from a handle at which it is still considered as hit.

Return Value

Returns an AdjustmentHandle instance representing the adjustment handle found at the specified location, or **null** if there is none.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
DrawHandles Method

DiagramItem.HitTestManipulators Method

Finds the **Manipulator** object located at the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Manipulator HitTestManipulators (
    Point point
)
```

Visual Basic

```
Public Overridable Function HitTestManipulators( _
    point As Point _
) As Manipulator
```

Parameters

point

Return Value

An instance of **Manipulator** -derived class, or **null** if there isn't one found at specified location.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.LabelFromPoint Method

Returns the ItemLabel located at specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public virtual ItemLabel LabelFromPoint (
    Point point
)
```

Visual Basic

```
Public Overridable Function LabelFromPoint( _
    point As Point _
) As ItemLabel
```

Parameters

point

Return Value

An ItemLabel instance.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
LinkLabel Class
NodeLabel Class

DiagramItem.LoadFromJson Method

Loads the item from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected virtual void LoadFromJson (
    JsonObject obj,
```

```
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject containing the item's data.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

Remarks

This method is invoked when an item is being deserialized from a diagram file saved in JSON format.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.LoadFromXml Method

Loads the item content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** containing the item's data.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

Remarks

This method is invoked when an item is being deserialized from a diagram file saved in XML format.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.LoadObsolete Method

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void LoadObsolete (
    XmlReader reader,
    SortedList brushes,
    Object extra
)
```

Visual Basic Copy Code

```
Protected Overridable Sub LoadObsolete( _
    reader As XmlReader, _
    brushes As SortedList, _
    extra As Object _
)
```

Parameters

reader

brushes

extra

Remarks

Invoked for control hosts when they are loaded from a pre-5.0 Xml file.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.NotInteractive Method

Gets whether the user is prevented from manipulating this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool NotInteractive ()
```

Visual Basic Copy Code

```
Public Overridable Function NotInteractive () As Boolean
```

Return Value

true to prevent user interaction with this item; otherwise, **false**.

Remarks

The default implementation prevents manipulation of invisible or locked items.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.NowModifying Method

Gets a flag indicating whether this object is being currently modified interactively by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool NowModifying ()
```

Visual Basic Copy Code

```
Public Function NowModifying () As Boolean
```

Return Value

true is the item is being modified; otherwise, **false**.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.OnAdd Method

Called when the item is added to a Diagram, either by the user, programmatically or via undo / redo.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnAdd ()
```

Visual Basic Copy Code

```
Protected Overridable Sub OnAdd ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
OnLoad Method

DiagramItem.OnChildModified Method

Called when a group child node has been modified.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnChildModified (
    DiagramNode node,
    AdjustmentHandles handle
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnChildModified( _
    node As DiagramNode, _
    handle As AdjustmentHandles _
)
```

Parameters

node
The node that has been modified.

handle
The adjustment handle used to modify the node.

Remarks

Override this method if you need to know when a node from the SubordinateGroup is modified.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.OnClick Method

Invoked when the user clicks this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnClick (
    MouseButton mouseButton,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnClick( _
    mouseButton As MouseButton, _
    mousePosition As Point _
)
```

Parameters

mouseButton
Specifies the mouse button used to click the item.

mousePosition
Specifies the mouse position.

Remarks

Implement this method in a custom item class if you need to detect clicks without handling events of the Diagram class.

See Also

DiagramItem Members
DiagramItem Class

DiagramItem.OnDoubleClick Method

Invoked when the user double-clicks this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnDoubleClick (
    MouseButton mouseButton,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnDoubleClick( _
    mouseButton As MouseButton, _
    mousePosition As Point _
)
```

Parameters

mouseButton
Specifies the mouse button used to click the item.

mousePosition
Specifies the mouse position.

Remarks

Implement this method in a custom item class if you need to detect double-clicks without handling events of the Diagram class.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.OnLoad Method

Called after successful deserialization.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void OnLoad ()
```

Visual Basic Copy Code

```
Public Overridable Sub OnLoad ()
```

Remarks

Override this method if you need to update the state of items after successfully loading a diagram.

See Also

DiagramItem.OnMeasureUnitChanged Method

Called when the diagram's MeasureUnit changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnMeasureUnitChanged ()
```

Visual Basic Copy Code

```
Protected Overridable Sub OnMeasureUnitChanged ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.OnRemove Method

Called when an item is removed from a Diagram, either by the user, programmatically or via undo / redo.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnRemove ()
```

Visual Basic Copy Code

```
Protected Overridable Sub OnRemove ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.OnStyleChanged Method

Invoked when the associated style or theme have changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnStyleChanged ()
```

Visual Basic Copy Code

```
Protected Overridable Sub OnStyleChanged ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.OnTextChanged Method

Called when the value of the Text property changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnTextChanged ()
```

Visual Basic Copy Code

```
Protected Overridable Sub OnTextChanged ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.OnTextFormatChanged Method

Called when the value of the TextFormat property changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnTextFormatChanged ()
```

Visual Basic Copy Code

```
Protected Overridable Sub OnTextFormatChanged ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.OnZoomChanged Method

Called when the diagram's **ZoomFactor** changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnZoomChanged ()
```

Visual Basic Copy Code

```
Protected Overridable Sub OnZoomChanged ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Query Method

Returns the items that can be reached from this item by following the path specified in the given query expression.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItemCollection Query (
    string query
)
```

Visual Basic Copy Code

```
Public Function Query( _
    query As String _
) As DiagramItemCollection
```

Parameters

query
A query expression consisting of selectors and boolean predicates.

Return Value

A collection of the found items.

Remarks

The query string consists of selectors, where each selector specifies the next items that should be selected starting from the current one.

The following selectors can be applied to a node:

links	Selects all links connected to the current node
inlinks	Selects the incoming links of the current node
outlinks	Selects the outgoing links of the current node
parent	Selects the parent item of the current node if it is in a group
children	Selects the child nodes of the current node if it is in a group

The following selectors can be applied to a link:

nodes	Selects the two nodes connected by the current link
origin	Selects the Origin node of the current link
destination	Selects the Destination node of the current link

children	Selects the child nodes of the current link if it is in a group.

Selectors must be separated by the / character. The results returned by a selector can be also filtered using a predicate expression, enclosed in square brackets after the selector name. The predicate expression can include C# like boolean and arithmetic operators, and one of the following property identifiers:

tag	Returns the value of the Tag property.
text	Returns the value of the Text property.
weight	Returns the value of the Weight property.

Example

The following code selects nodes two levels below the current one in a tree or layered graph.

C# Copy Code

```
DiagramItemCollection items = diagram.ActiveItem.Query(
    "outlinks/destination/outlinks/destination");
diagram.Selection.Clear();
foreach (DiagramItem item in items)
    item.Selected = true;
```

The following code finds the neighbors of the currently selected node that can be reached from it by following links with Weight greater than 3.

C# Copy Code

```
DiagramItemCollection items = diagram.ActiveItem.Query(
 "outlinks[weight>3]/destination");
```

The following example finds the destination nodes of links labeled "exception" going out of children of the current node's group.

C# Copy Code

```
DiagramItemCollection items = diagram.ActiveItem.Query(
    "children/outlinks[text==\"exception\"]/destination");
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
IncomingLinks Property
OutgoingLinks Property
MasterGroup Property
SubordinateGroup Property

DiagramItem.RemoveManipulator Method

Removes the specified manipulator from the object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void RemoveManipulator (
    Manipulator manipulator
)
```

Visual Basic Copy Code

```
Protected Sub RemoveManipulator( _
    manipulator As Manipulator _
)
```

Parameters

manipulator
An instance of **Manipulator** -derived class.

Remarks

Invoke this method to remove manipulators previously added using AddManipulator.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Repaint Method

Overload List

Name	Description
DiagramItem.Repaint ()	Repaints the region of the diagram where this item is located.
DiagramItem.Repaint (Boolean)	Repaints the specified region of the diagram.
DiagramItem.Repaint (Rect)	Repaints the specified region of the diagram.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Repaint Method ()

Repaints the region of the diagram where this item is located.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void Repaint ()
```

Visual Basic Copy Code

```
Protected Sub Repaint ()
```

Remarks

Call this method to repaint the diagram after changing appearance attributes of this item.

See Also

Repaint Method Overload List
DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Repaint Method (Boolean)

Repaints the specified region of the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void Repaint (
    bool includeConnected
)
```

Visual Basic Copy Code

```
Protected Sub Repaint( _
    includeConnected As Boolean _
)
```

Parameters

includeConnected
true if related items should be considered too; otherwise, **false**.

See Also

Repaint Method Overload List
DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Repaint Method (Rect)

Repaints the specified region of the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void Repaint (
    Rect invalidRect
)
```

Visual Basic Copy Code

```
Protected Sub Repaint( _
    invalidRect As Rect _
)
```

Parameters

invalidRect
The region to repaint.

Remarks

Call this method to repaint a part of the item after changing its appearance attributes.

See Also

Repaint Method Overload List
DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.RestoreItemProperties Method

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RestoreItemProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Public Sub RestoreItemProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.RestoreProperties Method

Restores the properties of this diagram item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void RestoreProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overridable Sub RestoreProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties -derived object containing a snapshot of the item's properties.

Remarks

This method is invoked by the undo/redo subsystem when undoing or redoing a ChangeItemCmd instance.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.RestoreState Method

Restores the state of this diagram item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void RestoreState (
    DiagramItemState state
)
```

Visual Basic Copy Code

```
Protected Overridable Sub RestoreState( _
    state As DiagramItemState _
)
```

Parameters

state
A DiagramItemState object.

Remarks

This method is called when an item should be restored to the state it had before user interaction.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
CreateState Method
SaveState Method

DiagramItem.SaveItemProperties Method

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveItemProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Public Sub SaveItemProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.SaveProperties Method

Saves the properties of this diagram item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SaveProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SaveProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties -derived object where the item's properties should be saved.

Remarks

This method is invoked by the undo/redo subsystem when saving a ChangeItemCmd instance.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.SaveState Method

Saves the state of this diagram item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SaveState (
    DiagramItemState state
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SaveState( _
    state As DiagramItemState _
)
```

Parameters

state
A DiagramItemState object.

Remarks

This method is called to save undo information when the users starts modifying an item.

See Also

DiagramItem Members
DiagramItem Class

DiagramItem.SaveToJson Method

Saves the item as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject instance to contain the serialized data.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

Remarks

This method is invoked when an item is being serialized into a JSON-format diagram file.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.SaveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** the item's data should be stored into.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

Remarks

This method is invoked when an item is being serialized into an XML-format diagram file.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.SetDiagramDirty Method

Marks the diagram as changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SetDiagramDirty ()
```

Visual Basic Copy Code

```
Protected Sub SetDiagramDirty ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.SetExpanded Method

Sets whether the item is in expanded or collapsed state.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SetExpanded (
    bool e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SetExpanded( _
    e As Boolean _
)
```

Parameters

e
Specifies the new state of the item.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.ShouldRender Method

Determines whether the item should be rendered in the current context.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected bool ShouldRender ()
```

Visual Basic Copy Code

```
Protected Function ShouldRender () As Boolean
```

Return Value

true to draw the item; otherwise, **false**.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.StartCreate Method

Invoked once when the user starts creating a new item interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void StartCreate (
    Point org
)
```

Visual Basic Copy Code

```
Protected Overridable Sub StartCreate( _
    org As Point _
)
```

Parameters

org
Position of the mouse pointer.

Remarks

This method is invoked in response to the **MouseDown** event, just after an item has been instantiated.

See Also

DiagramItem.StartModify Method

Invoked when the user starts modifying this item interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void StartModify (
    Point org,
    AdjustmentHandle handle,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overridable Sub StartModify( _
    org As Point, _
    handle As AdjustmentHandle, _
    ist As InteractionState _
)
```

Parameters

org
Position of the mouse pointer.

handle
The adjustment handle used to modify this item.

ist
An InteractionState object containing more information about the current state of the operation.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.UpdateCreate Method

Invoked during interactive creation of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void UpdateCreate (
    Point current
)
```

Visual Basic Copy Code

```
Protected Overridable Sub UpdateCreate( _
    current As Point _
)
```

Parameters

current
Position of the mouse pointer.

Remarks

This methods is invoked after each mouse movement in order to update the item's state in accordance with the new mouse position.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.UpdateGroup Method

Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void UpdateGroup ()
```

Visual Basic Copy Code

```
Protected Overridable Sub UpdateGroup ()
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Group.AttachCustom Method

DiagramItem.UpdateModify Method

Invoked during interactive modification of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void UpdateModify (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overridable Sub UpdateModify( _
    current As Point, _
    ist As InteractionState _
)
```

Parameters

current
Position of the mouse pointer.

ist
An InteractionState object containing more information about the current state of the operation.

Remarks

This methods is invoked after each mouse movement in order to update the item's state in accordance with the new mouse position.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.ZBottom Method

Moves the item to the bottom of the z-order.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ZBottom (
    bool keepInLayer
)
```

Visual Basic Copy Code

```
Public Sub ZBottom( _
    keepInLayer As Boolean _
)
```

Parameters

keepInLayer

Remarks

Call the method to move the item to the bottom of the z-order.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
ZTop Method
ZIndex Property

DiagramItem.ZLevelDown Method

Moves the item one level down in the z-order.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ZLevelDown (
    bool keepInLayer
)
```

Visual Basic Copy Code

```
Public Sub ZLevelDown( _
    keepInLayer As Boolean _
```

```
)
```

Parameters

keepInLayer

Remarks

Call the method to change the item's position in z-order with one level down.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
ZLevelUp Method
ZIndex Property

DiagramItem.ZLevelUp Method

Moves the item one level up in the z-order.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ZLevelUp (
    bool keepInLayer
)
```

Visual Basic Copy Code

```
Public Sub ZLevelUp( _
    keepInLayer As Boolean _
)
```

Parameters

keepInLayer

Remarks

Call the method to change the item's position in z-order with one level up.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
ZLevelDown Method
ZIndex Property

DiagramItem.ZTop Method

Moves the item to the top of the z-order.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ZTop (
    bool keepInLayer
```

```
)
```

Visual Basic Copy Code

```
Public Sub ZTop( _
    keepInLayer As Boolean _
)
```

Parameters

keepInLayer

Remarks

Call the method to move the item to the top of the z-order.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
ZBottom Method
ZIndex Property

DiagramItem Properties

DiagramItem.Brush Property

Gets or sets the brush used to fill the interior of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush Brush { get; set; }
```

Visual Basic Copy Code

```
Public Property Brush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class. The default value is **null**.

Remarks

This property refers to a brush with which an item is painted. You can assign to it an instance of one of the following classes: HatchBrush, LinearGradientBrush, PathGradientBrush, SolidBrush or TextureBrush, located in the MindFusion.Drawing namespace. When applied to links, the property defines how arrowheads interior is painted.

When set to **null**, the brush object used to paint this item is derived from the item's Style, or from the diagram's Theme.

Example

The following example demonstrates how to create a **LinearGradientBrush** and assign it to the first node of a diagram:

C# Copy Code

```
diagram.Nodes[0].Brush = new MindFusion.Drawing.LinearGradientBrush(
    Color.Yellow, Color.BurlyWood, 20);
```

Visual Basic Copy Code

```
Diagram.Nodes(0).Brush = New MindFusion.Drawing.LinearGradientBrush( _
    Color.Yellow, Color.BurlyWood, 20)
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
EffectiveBrush Property
Opacity Property

DiagramItem.CssClass Property

Gets or sets the CSS class associated with this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string CssClass { get; set; }
```

Visual Basic Copy Code

```
Public Property CssClass As String
```

Property Value

A string containing the CSS class.

Remarks

At this time **CssClass** is considered only by ASP.NET MVC version of the library.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.EffectiveBrush Property

Gets the effective brush of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush EffectiveBrush { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectiveBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

Remarks

Returns a Brush instance used to paint this item. This value comes from the Brush property if it is set, or is derived from either the item's Style or the current diagram Theme.

See Also

DiagramItem.EffectiveFont Property

Gets the currently effective font of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Font EffectiveFont { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectiveFont As Font
```

Property Value

A Font instance.

Remarks

This property returns Font if it is not **null**, or a font with the FontFamily, FontSize, FontStyle and FontUnit properties of the item's style.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.EffectivePen Property

Gets the effective pen of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen EffectivePen { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectivePen As Pen
```

Property Value

An instance of the MindFusion.Drawing.Pen class.

Remarks

Returns a Pen instance used to draw this item's border or segment lines. This value comes from the Pen property if it is set, or is derived from either the item's Style or the current diagram Theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.EffectiveShadowBrush Property

Gets the effective shadow brush of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush EffectiveShadowBrush { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectiveShadowBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

Remarks

Returns a Brush instance used to paint the item's shadow. This value comes from the ShadowBrush property if it is set, or is derived from either the item's Style or the current diagram Theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.EffectiveTextBrush Property

Gets the effective text brush of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush EffectiveTextBrush { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectiveTextBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

Remarks

Returns a Brush instance used to paint the item's text. This value comes from the TextBrush property if it is set, or is derived from either the item's Style or the current diagram Theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Font Property

Gets or sets the font used to render the item's text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Font Font { get; set; }
```

Visual Basic Copy Code

```
Public Overridable Property Font As Font
```

Property Value

An instance of the Font class. The default value is **null**.

Remarks

Items derived from DiagramItem use this font to draw their text. Tables for example render their captions and text in cells using this font. Initially the font is set to the value of Font.

When set to **null**, the font object used to render the item's text is derived from the item's Style, or from the diagram's current Theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
EffectiveFont Property
Text Property
TextBrush Property

DiagramItem.HyperLink Property

Gets or sets the hyperlink associated with this diagram item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string HyperLink { get; set; }
```

Visual Basic Copy Code

```
Public Property HyperLink As String
```

Property Value

A string value containing the hyperlink URL.

Remarks

This property is used by the CreateImageMap method of the HtmlBuilder class. Its value is assigned to the HREF attribute of the <AREA> tag that corresponds to the diagram item.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Id Property

Gets or sets user-defined identifier associated with this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As Object
```

Property Value

An instance of any class derived from **System.Object**.

Remarks

This property contains data that you define and is not used by the component in any way. You can associate with the item any value of a primitive data type, an object from the .NET class library or a custom class.

If the id is a value of primitive type, it is serialized directly by the component along with item containing it when saving / loading the diagram. Reference types are serialized using **System.Xml.Serialization.XmlSerializer** when saving diagrams to XML, or using **System.Text.Json.JsonSerializer** when saving to JSON.

You can implement custom serialization code using .NET XML API by handling SerializeTag and DeserializeTag events, or Json API using JsonSerializeTag and JsonDeserializeTag events

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Diagram.FindNodeById Method
Diagram.FindLinkById Method
Tag Property

DiagramItem.IgnoreLayout Property

Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IgnoreLayout { get; set; }
```

Visual Basic Copy Code

```
Public Property IgnoreLayout As Boolean
```

Property Value

true if the item must stay at its place nevertheless an automatic layout is performed; otherwise, **false**. The default is **false**.

Remarks

Enable **IgnoreLayout** to specify that the Arrange method of layout classes should ignore an item. If enabled, the position of an item won't be changed. This property applies to visible nodes. Invisible nodes are always ignored by automatic layout methods.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Locked Property
Visible Property

DiagramItem.LayerIndex Property

Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int LayerIndex { get; set; }
```

Visual Basic Copy Code

```
Public Property LayerIndex As Integer
```

Property Value

An integer value that specifies the index of the layer.

Remarks

This property specifies the index of a Layer instance within the Layers collection of the Diagram.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
ZIndex Property

DiagramItem.LayoutTraits Property

Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IDictionary LayoutTraits { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property LayoutTraits As IDictionary
```

Property Value

A hashtable of key-value pairs that define layout-specific properties for this item.

Remarks

This property is used by some Layout classes and the DiagramView3D class:

- **FlowLayout** looks for values assigned to keys from the FlowLayoutTraits enumeration.
- SpringLayout looks for values associated with SpringLayoutTraits members.
- TreeLayout looks for values associated with TreeLayoutTraits members.
- SwimlaneLayout looks for values associated with SwimlaneLayoutTraits members.
- LayeredLayout looks for values associated with LayeredLayoutTraits members.
- HierarchicalLayout looks for values associated with HierarchicalLayoutTraits members.

- The **DiagramView3D** class seeks keys values from the View3D enumeration.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Locked Property

Gets or sets a value indicating whether users are allowed to edit the diagram item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Locked { get; set; }
```

Visual Basic Copy Code

```
Public Property Locked As Boolean
```

Property Value

true to prevent users from editing a diagram element; otherwise, **false**. The default is **false**.

Remarks

A locked item cannot be selected, moved or resized interactively. Additionally, its text cannot be edited in-place.

Locked nodes still can be moved or resized programmatically through the Bounds property and the Move / Resize methods. Positions of locked links' control points can be changed programmatically too - through their ControlPoints collections.

Locked items positions can be changed also by the automatic layout algorithms. To prevent automatic layout methods from changing positions of individual items, set the IgnoreLayout property of DiagramItem to **false**.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Visible Property

DiagramItem.MeasureUnit Property

Returns the diagram measure unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected MeasureUnit MeasureUnit { get; }
```

Visual Basic Copy Code

```
Protected ReadOnly Property MeasureUnit As MeasureUnit
```

Property Value

An instance of the MeasureUnit class.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Opacity Property

Gets or sets the opacity of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Opacity { get; set; }
```

Visual Basic Copy Code

```
Public Property Opacity As Double
```

Property Value

A value between 0 and 1 specifying the item's opacity.

Remarks

This value is applied as a factor multiplying the alpha channel of all colors in the item's graphics.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Brush Property
Pen Property

DiagramItem.Parent Property

Gets the diagram that contains this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Diagram Parent { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Parent As Diagram
```

Property Value

A reference to a Diagram instance.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Pen Property

Gets or sets the pen used to paint the frame of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen Pen { get; set; }
```

Visual Basic Copy Code

```
Public Property Pen As Pen
```

Property Value

An instance of the MindFusion.Drawing.Pen class. The default value is **null**.

Remarks

The property refers to a pen, with which the item frame is painted. When applied to links, the property specifies how their segments are painted.

When set to **null**, the pen object used to draw the item borders is derived from the item's Style, or from the diagram's current Theme.

Example

The following example demonstrates how to create a pen with a linear gradient and assign it to the first node of a diagram:

C# Copy Code

```
MindFusion.Drawing.Brush penBrush = new MindFusion.Drawing.LinearGradientBrush(
    Color.Bisque, Color.Blue, 0);
diagram.Nodes[0].Pen = new MindFusion.Drawing.Pen(penBrush);
```

Visual Basic Copy Code

```
Dim penBrush As New MindFusion.Drawing.LinearGradientBrush( _
    Color.Bisque, Color.Blue, 0)
diagram.Nodes(0).Pen = New MindFusion.Drawing.Pen(penBrush)
```

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
EffectivePen Property
Opacity Property

DiagramItem.Printable Property

Gets or sets a value indicating whether to print the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Printable { get; set; }
```

Visual Basic Copy Code

```
Public Property Printable As Boolean
```

Property Value

true to print this item; otherwise, **false**.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Visible Property
Diagram.Print Method

DiagramItem.RuntimeIndex Property

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int RuntimeIndex { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property RuntimeIndex As Integer
```

Property Value

A sequential integer id for this item.

Remarks

Valid only after calling UpdateRuntimeIndices.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Selected Property

Gets or sets a value indicating whether a diagram item is selected.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Selected { get; set; }
```

Visual Basic Copy Code

```
Public Property Selected As Boolean
```

Property Value

true if the item is selected; otherwise, **false**.

Remarks

Use the property to check which items are selected by the user, or to programmatically select items if needed. Selected items adjustment handles are painted using the pen and brush specified through the SelectedItemHandlesStyle property. You can specify if users first have to select items in order to modify them with the **ModificationStart** property.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Selection Class
Diagram.Selection Property

DiagramItem.ShadowBrush Property

Gets or sets the brush used to paint the item's shadow.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush ShadowBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property ShadowBrush As Brush
```

Property Value

An instance of a Brush -derived class. The default value is **null**.

Remarks

The shadow of an item has the same shape as the item, but is painted with a different brush and with some offset from the item position. This property specifies the brush used to paint the item's shadow.

When set to **null**, the brush object used to paint the shadow is derived from the item's Style, or from the current diagram Theme.

Example

The following code creates a shadow with fuzzy edges for the newly created node.

C# Copy Code

```
private void diagram_InitializeNode(object sender, NodeEventArgs e)
{
    ColorBlend blend = new ColorBlend(3);
    blend.Colors = new Color[] {
        Color.Transparent,
        Color.FromArgb(128, Color.Black),
        Color.FromArgb(128, Color.Black) };
    blend.Positions = new float[] { 0f, 0.1f, 1f };

    PathGradientBrush brush = new PathGradientBrush(Color.White, Color.White);
    brush.InterpolationColors = blend;

    e.Node.ShadowBrush = brush;
    e.Node.ShadowOffsetY = 2;
    e.Node.ShadowOffsetX = 1.5f;
}
```



See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Diagram.ShadowsStyle Property
EffectiveShadowBrush Property

DiagramItem.ShadowOffsetX Property

Gets or sets the horizontal offset of the shadow of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ShadowOffsetX { get; set; }
```

Visual Basic Copy Code

```
Public Property ShadowOffsetX As Double
```

Property Value

A **double** value specifying the horizontal offset of this item's shadow. Initialized with the value of ShadowOffsetX.

Remarks

The shadow of an item has the same shape as the item itself, but with a different color and with some offset from the item position. This property specifies the horizontal offset of the shadow.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
ShadowOffsetY Property
Diagram.ShadowsStyle Property

DiagramItem.ShadowOffsetY Property

Gets or sets the vertical offset of the shadow of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ShadowOffsetY { get; set; }
```

Visual Basic Copy Code

```
Public Property ShadowOffsetY As Double
```

Property Value

A **double** value specifying the vertical offset of this item's shadow. Initialized with the value of ShadowOffsetY.

Remarks

The shadow of an item has the same shape as the item itself, but with a different color and with some offset from the item position. This property specifies the vertical offset of the shadow.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
ShadowOffsetX Property
Diagram.ShadowsStyle Property

DiagramItem.Style Property

Gets or sets the style associated with the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItemStyle Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As DiagramItemStyle
```

Property Value

An instance of a DiagramItemStyle-derived class.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.SubordinateGroup Property

Gets a reference to the group of items that are attached to this one.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Group SubordinateGroup { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property SubordinateGroup As Group
```

Property Value

A reference to a Group object or **null** (**Nothing** in Visual Basic) if the item is not master of a group.

Remarks

Diagram items can be grouped hierarchically in such a manner, that if an item is moved all of its subordinated items in the hierarchy move too. The immediate subordinates of an item are contained and represented by a Group instance. **SubordinateGroup** provides access to the group of items that are attached to this one. If there are no attached items, the property returns **null** (**Nothing** in Visual Basic).

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
MasterGroup Property
Factory.CreateGroup Method
Query Method

DiagramItem.Tag Property

Gets or sets user-defined data associated with this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Tag { get; set; }
```

Visual Basic Copy Code

```
Public Property Tag As Object
```

Property Value

An instance of any class derived from **System.Object**.

Remarks

This property contains data that you define and is not used by the component in any way. You can associate with the item any value of a primitive data type, an object from the .NET class library or a custom class.

If the tag is a value of primitive type, it is serialized directly by the component along with item containing it when saving / loading the diagram. Reference types are serialized using **System.Xml.Serialization.XmlSerializer** when saving diagrams to XML, or using **System.Text.Json.JsonSerializer** when saving to JSON.

You can implement custom serialization code using .NET XML API by handling SerializeTag and DeserializeTag events, or Json API using JsonSerializeTag and JsonDeserializeTag events

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Diagram.FindNode Method
Diagram.FindLink Method
Id Property

DiagramItem.Text Property

Gets or sets the text displayed by this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Text { get; set; }
```

Visual Basic Copy Code

```
Public Property Text As String
```

Property Value

A string containing the item's text. The default value is empty string "".

Remarks

This property defines item's main text content. You can add more text labels by calling the AddLabel method, and customize their position and appearance via the various members of returned ItemLabel object.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Font Property

TextFormat Property
TextBrush Property

DiagramItem.TextBrush Property

Gets or sets the brush used to paint texts inside the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush TextBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property TextBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class. The default value is **null**.

Remarks

When set to **null**, the brush object used to draw text is derived from either the item's Style or the current diagram Theme.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
EffectiveTextBrush Property

DiagramItem.TextFormat Property

Gets or sets the alignment of text displayed inside this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringFormat TextFormat { get; set; }
```

Visual Basic Copy Code

```
Public Property TextFormat As StringFormat
```

Property Value

A StringFormat instance.

Remarks

Set this property to specify the wrapping mode and alignment of item's Text.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Text Property
Font Property

DiagramItem.TextPadding Property

Gets or sets the spacing between the item boundaries and its text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Thickness TextPadding { get; set; }
```

Visual Basic Copy Code

```
Public Overridable Property TextPadding As Thickness
```

Property Value

A Thickness instance specifying the size of the padding space left around the text.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.ToolTip Property

Gets or sets a short text message displayed when the mouse pointer rests over the diagram element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ToolTip { get; set; }
```

Visual Basic Copy Code

```
Public Property ToolTip As String
```

Property Value

The tool-tip text displayed for this item. The default value is an empty string.

Remarks

When the mouse rests for a while over an item, the item's tool-tip text is displayed. If it is an empty string, the diagram's ToolTip is displayed instead.

The tool-tip text is also used as a value of the ALT attribute of the image map AREA corresponding to the item; see HtmlBuilder for further information.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
DiagramView.ShowToolTips Property
DiagramView.ToolTipDelay Property

DiagramItem.TopLevel Property

True if this is a top-level item in the diagram, or false if it is inside a container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool TopLevel { get; }
```

Visual Basic Copy Code

```
Public Overridable ReadOnly Property TopLevel As Boolean
```

Property Value

true if this is a top-level item in the diagram; **false** otherwise.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.Visible Property

Gets or sets a value indicating whether this item is visible.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool Visible { get; set; }
```

Visual Basic Copy Code

```
Public Overridable Property Visible As Boolean
```

Property Value

true if the item is visible; otherwise, **false**.

Remarks

Invisible items are not drawn at all. You might use the property to hide or show items temporarily, or to construct complex group hierarchies in which intermediate helper items are invisible.

Invisible nodes are ignored by automatic layout methods. The position of such nodes won't change when the Arrange method of a layout class is called. Visible items can be ignored too by enabling their IgnoreLayout property.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
Printable Property
Locked Property

DiagramItem.Weight Property

Gets or sets a weight value used in layout and path-finding algorithms.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Weight { get; set; }
```

Visual Basic Copy Code

```
Public Property Weight As Double
```

Property Value

A **double** value specifying the weight of this item. The default value is 1.

Remarks

Currently **Weight** is used in the implementations of:

- The SpringLayout class: the weight of a link connecting two nodes is multiplied by the desired NodeDistance to determine the final distance between the nodes.
- The PathFinder class: an override of the FindShortestPath method uses the **Weight** values of nodes and/or links to find the path with smallest total weight.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace

DiagramItem.ZIndex Property

Gets or sets the z-order position of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int ZIndex { get; set; }
```

Visual Basic Copy Code

```
Public Property ZIndex As Integer
```

Property Value

An integer value specifying the position of the diagram item within the z-order. The default value is -1.

Remarks

Use this property to arrange items in z-order. Items with lower **ZIndex** appear behind items with higher **ZIndex**.

If the ZIndex property of newly added items is equal to -1, the control will automatically assign consecutively increasing values to it. If you set ZIndex to a value greater then -1 before adding an item, the control will preserve it.

See Also

DiagramItem Members
DiagramItem Class
MindFusion.Diagramming Namespace
ZLevelDown Method
ZBottom Method
ZLevelUp Method
ZTop Method
LayerIndex Property

DiagramItemCollection Class

Represents a strongly typed collection of DiagramItem objects.

Syntax

C# Copy Code

```
public class DiagramItemCollection : ItemCollectionBase<DiagramItem>
```

Visual Basic Copy Code

```
Public Class DiagramItemCollection
    Inherits ItemCollectionBase(Of DiagramItem)
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.CollectionBase(DiagramItem)
    MindFusion.Diagramming.ItemCollectionBase(DiagramItem)
      MindFusion.Diagramming.DiagramItemCollection

See Also

DiagramItemCollection Members
MindFusion.Diagramming Namespace

DiagramItemCollection Members

Represents a strongly typed collection of DiagramItem objects.

The following tables list the members exposed by the DiagramItemCollection type.

Public Constructors

	Name	Description
	DiagramItemCollection	Initializes a new instance of the **DiagramItemCollection** class.

Protected Fields

	Name	Description
	InnerList	Gets a **List<T>** containing the list of elements in the collection instance. (Inherited from CollectionBase.)
	itemEventArgs	A ItemEventArgs instance used for passing arguments to the event handlers for the events of this collection. Used to avoid creating many small objects that would burden the garbage collector. (Inherited from ItemCollectionBase.)
	notifyParent	If enabled, the collection raises events to let their containing objects handle the addition or removal of items. (Inherited from ItemCollectionBase.)

Public Properties

	Name	Description
	Count	Gets the number of elements actually contained in the collection. (Inherited from CollectionBase.)
	IsReadOnly	Gets a value indicating whether the collection is read-only. (Inherited from CollectionBase.)
	Item	Gets or sets the element at the specified index. (Inherited from CollectionBase.)

Public Methods

	Name	Description
	Add	Adds the specified diagram item to the collection. (Inherited from ItemCollectionBase.)
	Clear	Removes all diagram items from the collection. (Inherited from ItemCollectionBase.)
	Clone	Creates a shallow copy of this collection.
	Contains	Determines whether an element is in the collection. (Inherited from CollectionBase.)
	CopyTo	Copies the entire collection to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from CollectionBase.)
	GetAt	Gets the diagram item at the specified position. (Inherited from ItemCollectionBase.)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from CollectionBase.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire collection. (Inherited from CollectionBase.)
	Insert	Inserts the specified diagram item into the collection at the specified position. (Inherited from ItemCollectionBase.)
	Remove	Removes the diagram item from the collection. (Inherited from ItemCollectionBase.)
	RemoveAt	Removes the diagram item from the collection. (Inherited from ItemCollectionBase.)
	Sort	Overloaded. (Inherited from CollectionBase.)

Protected Methods

	Name	Description
	RaiseAdding	Raises the **Adding** event. (Inherited from ItemCollectionBase.)
	RaiseInserting	Raises the **Inserting** event. (Inherited from ItemCollectionBase.)
	RaiseRemoving	Raises the **Removing** event. (Inherited from ItemCollectionBase.)

See Also

DiagramItemCollection Class
MindFusion.Diagramming Namespace

DiagramItemCollection Constructor

Initializes a new instance of the DiagramItemCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItemCollection ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramItemCollection Class
MindFusion.Diagramming Namespace

DiagramItemCollection Methods

DiagramItemCollection.Clone Method

Creates a shallow copy of this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItemCollection Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As DiagramItemCollection
```

Return Value

An instance of the DiagramItemCollection class.

See Also

DiagramItemCollection Members
DiagramItemCollection Class
MindFusion.Diagramming Namespace

DiagramItemProperties Class

The base type for classes that store diagram item properties during undo / redo operations.

Syntax

C# Copy Code

```
public class DiagramItemProperties
```

Visual Basic Copy Code

```
Public Class DiagramItemProperties
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItemProperties
    MindFusion.Diagramming.DiagramLinkProperties
    MindFusion.Diagramming.DiagramNodeProperties

See Also

DiagramItemProperties Members
MindFusion.Diagramming Namespace

DiagramItemProperties Members

The base type for classes that store diagram item properties during undo / redo operations.

The following tables list the members exposed by the DiagramItemProperties type.

Public Constructors

	Name	Description
	DiagramItemProperties	

Public Fields

	Name	Description
	ItemClass	For internal use.
	Selected	For internal use.
	Text	For internal use.
	TextPadding	For internal use.
	ZIndex	For internal use.

Public Methods

	Name	Description
	FreeResources	Frees stored resources which will no longer be used.

See Also

DiagramItemProperties Class
MindFusion.Diagramming Namespace

DiagramItemProperties Fields

DiagramItemProperties.ItemClass Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ItemClass
```

Visual Basic Copy Code

```
Public ItemClass As String
```

See Also

DiagramItemProperties Members
DiagramItemProperties Class
MindFusion.Diagramming Namespace

DiagramItemProperties.Selected Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Selected
```

Visual Basic Copy Code

```
Public Selected As Boolean
```

See Also

DiagramItemProperties Members
DiagramItemProperties Class
MindFusion.Diagramming Namespace

DiagramItemProperties.Text Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Text
```

Visual Basic Copy Code

```
Public Text As String
```

See Also

DiagramItemProperties Members
DiagramItemProperties Class
MindFusion.Diagramming Namespace

DiagramItemProperties.TextPadding Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Thickness TextPadding
```

Visual Basic Copy Code

```
Public TextPadding As Thickness
```

See Also

DiagramItemProperties Members
DiagramItemProperties Class
MindFusion.Diagramming Namespace

DiagramItemProperties.ZIndex Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int ZIndex
```

Visual Basic Copy Code

```
Public ZIndex As Integer
```

See Also

DiagramItemProperties Members
DiagramItemProperties Class
MindFusion.Diagramming Namespace

DiagramItemProperties Constructor

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItemProperties ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramItemProperties Class
MindFusion.Diagramming Namespace

DiagramItemProperties Methods

DiagramItemProperties.FreeResources Method

Frees stored resources which will no longer be used.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void FreeResources ()
```

Visual Basic Copy Code

```
Public Overridable Sub FreeResources ()
```

See Also

DiagramItemProperties Members
DiagramItemProperties Class
MindFusion.Diagramming Namespace

DiagramItemState Class

Contains a snapshot of the internal state of a diagram item. Instances of this class are used to store the state of items for later undo or redo operations.

Syntax

C# Copy Code

```
public class DiagramItemState
```

Visual Basic Copy Code

```
Public Class DiagramItemState
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItemState
    MindFusion.Diagramming.DiagramLinkState
    MindFusion.Diagramming.DiagramNodeState

See Also

DiagramItemState Members
MindFusion.Diagramming Namespace

DiagramItemState Members

Contains a snapshot of the internal state of a diagram item. Instances of this class are used to store the state of items for later undo or redo operations.

The following tables list the members exposed by the DiagramItemState type.

Public Constructors

	Name	Description
	DiagramItemState	Initializes a new instance of the **DiagramItemState** class.

See Also

DiagramItemState Class
MindFusion.Diagramming Namespace

DiagramItemState Constructor

Initializes a new instance of the DiagramItemState class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItemState ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramItemState Class
MindFusion.Diagramming Namespace

DiagramItemStyle Class

Represents a reusable set of properties that define the appearance of diagram items.

Syntax

C# Copy Code

```
public class DiagramItemStyle : Style
```

Visual Basic Copy Code

```
Public Class DiagramItemStyle
    Inherits Style
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle
      MindFusion.Diagramming.DiagramLinkStyle
      MindFusion.Diagramming.DiagramNodeStyle

See Also

DiagramItemStyle Members
MindFusion.Diagramming Namespace

DiagramItemStyle Members

Represents a reusable set of properties that define the appearance of diagram items.

The following tables list the members exposed by the DiagramItemStyle type.

Protected Constructors

	Name	Description
	DiagramItemStyle	Initializes a new instance of the **DiagramItemStyle** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of the item.

	FontFamily	Gets or sets the font family of the item text.
	FontSize	Gets or sets the font size of the item text.
	FontStyle	Gets or sets the font style of the item text.
	FontUnit	Gets or sets the font unit of the item text.
	ShadowBrush	Gets or sets the brush used to paint the shadow of the item.
	Stroke	Gets or sets the brush used to paint the outline of the item.
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items.
	StrokeThickness	Gets or sets the thickness of the item's outline, expressed in the current units of measure.
	TextBrush	Gets or sets the brush used to paint texts inside the item.

Public Methods

	Name	Description
	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
S	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JsonObject.
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)
	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a **JsonObject**.
	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle Constructor

Initializes a new instance of the DiagramItemStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected DiagramItemStyle ()
```

Visual Basic Copy Code

```
Protected Sub New ()
```

See Also

DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle Methods

DiagramItemStyle.LoadFromJson Method

Loads the style content from a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the style's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle.SaveToJson Method

Saves the style content into a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject the style's data should be stored into.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle Properties

DiagramItemStyle.Brush Property

Gets or sets the brush used to fill the interior of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush Brush { get; set; }
```

Visual Basic Copy Code

```
Public Property Brush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle.FontFamily Property

Gets or sets the font family of the item text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string FontFamily { get; set; }
```

Visual Basic Copy Code

```
Public Property FontFamily As String
```

Property Value

A string value.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle.FontSize Property

Gets or sets the font size of the item text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double FontSize { get; set; }
```

Visual Basic Copy Code

```
Public Property FontSize As Double
```

Property Value

A **double** value.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle.FontStyle Property

Gets or sets the font style of the item text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public FontStyle FontStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property FontStyle As FontStyle
```

Property Value

One or more of the members of the **FontStyle** enumeration.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle.FontUnit Property

Gets or sets the font unit of the item text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public GraphicsUnit FontUnit { get; set; }
```

Visual Basic Copy Code

```
Public Property FontUnit As GraphicsUnit
```

Property Value

A member of the **GraphicsUnit** enumeration.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle.ShadowBrush Property

Gets or sets the brush used to paint the shadow of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush ShadowBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property ShadowBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle.Stroke Property

Gets or sets the brush used to paint the outline of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush Stroke { get; set; }
```

Visual Basic Copy Code

```
Public Property Stroke As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle.StrokeDashStyle Property

Gets or sets the pattern of dashes and gaps used to outline items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DashStyle StrokeDashStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property StrokeDashStyle As DashStyle
```

Property Value

A member of the **DashStyle** enumeration.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle.StrokeThickness Property

Gets or sets the thickness of the item's outline, expressed in the current units of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double StrokeThickness { get; set; }
```

Visual Basic

```
Public Property StrokeThickness As Double
```

Property Value

A double value.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle.TextBrush Property

Gets or sets the brush used to paint texts inside the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Brush TextBrush { get; set; }
```

Visual Basic

```
Public Property TextBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

DiagramItemStyle Members
DiagramItemStyle Class
MindFusion.Diagramming Namespace

DiagramItemStyle Events

DiagramLink Class

Represents a link between two diagram nodes.

Syntax

C#

```
public class DiagramLink : DiagramItem, Routable, IAnimationItem
```

Visual Basic

```
Public Class DiagramLink
    Inherits DiagramItem
    Implements Routable, IAnimationItem
```

Remarks

Links between nodes in MvcDiagram diagrams are represented by instances of the **DiagramLink** class. Links connect at most two nodes, which can be accessed via the Origin and Destination properties. If connected to node anchor points, the point indices can be accessed using the OriginAnchor and DestinationAnchor properties. To control whether users are allowed to move link end points, set AllowMoveStart and AllowMoveEnd.

Links are built of smaller parts called segments, which can be straight lines, cascading orthogonal lines, Bézier or spline curves. The type of segments is set through the Shape property. The number of link's segments is specified by the SegmentCount property. Access to control point locations is provided by the ControlPoints collection.

Typically, only the start and end segments of a link are updated when the nodes it connects are moved. Number, orientation and/or position of link segments can change dynamically in various ways depending on the values of AutoRoute and Dynamic properties.

If AutoSnapToNode is enabled, a link automatically aligns its end points to the border of the origin or destination nodes when a user moves the first or last control points. This has a higher priority than aligning to the grid, but a lower priority than aligning to an anchor point.

Links can display distinct arrowhead shapes at their ends. These shapes are configured using the HeadShape and BaseShape properties. The sizes of the base and arrowhead images are specified with the HeadShapeSize and BaseShapeSize properties. Arrowhead shapes can also be displayed at the midpoint of each segment to help distinguish between different links. The type and size of these shapes are controlled by the IntermediateShape and IntermediateShapeSize properties. The outlines of the arrowhead shapes are rendered using the HeadPen pen.

The color of link segments is specified by the Pen property, while the fill color of the base and head shapes is controlled by the BaseBrush and HeadBrush properties. The appearance of link's selection handles can be changed using HandlesStyle.

Links can have associated text, displayed according to the Text, TextBrush and TextStyle properties. The AddLabel method allows multiple labels to be displayed per link. Labels can automatically avoid overlapping nodes or other labels when their AutoArrange property is enabled.

Bounds and Length are informative properties returning respectively link's location and total length.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramLink

See Also

DiagramLink Members
MindFusion.Diagramming Namespace
LinkLabel Class
DiagramNode Class

DiagramLink Members

Represents a link between two diagram nodes.

The following tables list the members exposed by the DiagramLink type.

Public Constructors

	Name	Description
	DiagramLink	Overloaded. Initializes a new instance of the **DiagramLink** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowMoveEnd	Gets or sets a value indicating whether the user is allowed to move the end point of a link.
	AllowMoveStart	Gets or sets a value indicating whether the user is allowed to move the first control point of a link.
	AutoRoute	Gets or sets a value indicating whether a link should avoid nodes by going the shortest path from its origin to its destination without crossing any other nodes.

	AutoSnapToNode	Specifies whether a link end point should automatically snap to the nearest diagram node while the user moves the point's adjustment handle.
	BaseBrush	Gets or sets a brush used to fill the arrowhead drawn at the link base.
	BaseShape	Gets or sets what shape to display at the beginning of a link.
	BaseShapeSize	Gets or sets the size of the **BaseShape** shape.
	Bounds	Gets the size of the rectangle that bounds this link.
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	CascadeOrientation	Gets or sets the orientation of the first segment of a cascading link.
	ControlPoints	Gets a collection of the control points of this link.
	CrossingRadius	Gets or sets the radius length of decorations displayed at intersection points of this link.
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	CustomDraw	Gets or sets the type of custom drawing performed on this link.
	Destination	Gets or sets the destination node of this link.
	DestinationAnchor	Gets or sets the index of the destination anchor point within the AnchorPattern that contains it.
	DestinationConnection	Gets or sets the ConnectionPoint object representing the link connection to its destination node. This allows connecting the link to a constituent part of a node, such as a TableNode row or a TreeViewItem.
	DestinationIndex	Gets or sets the index of link's destination anchor point or table row.
	DrawCrossings	Gets or sets a value indicating whether the points where this link crosses other links should be drawn as arcs or dashes.
	Dynamic	Gets or sets a value indicating whether a link automatically adjusts its end points positions in order to keep pointing to the centers of nodes that it connects.
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectiveHeadPen	Gets the effective head pen of the link.
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	EndPoint	Gets or sets the last control point of a link.
	Font	DiagramItem.Font override.
	HandlesStyle	Gets or sets the appearance of the link's selection handles.
	HeadBrush	Gets or sets a brush used to fill the arrowhead drawn at the link's destination.
	HeadPen	Gets or sets the pen to use when painting arrowheads lines.
	HeadShape	Gets or sets the shape of this link's arrowhead.
	HeadShapeSize	Gets or sets the size of the arrowhead shape.
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)

	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IntermediateBrush	Gets or sets a brush used to fill the arrowheads drawn at the middle of link segments.
	IntermediateShape	Gets or sets the shape displayed at the middle of each segment of the link.
	IntermediateShapeSize	Gets or sets the size of the **IntermediateShape** shape.
	IsConnected	Gets a value indicating whether the link is connected to nodes at both its ends.
	Labels	Returns the list of all labels displayed by this link.
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Length	Gets the length of the link.
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	Origin	Gets or sets the origin node of the link.
	OriginAnchor	Gets or sets the index of the origin anchor point within the **AnchorPattern** that contains it.
	OriginConnection	Gets or sets the **ConnectionPoint** object representing the link connection to its origin node. This allows connecting the link to a constituent part of a node, such as a **TableNode** row or a **TreeViewItem**.
	OriginIndex	Gets or sets the index of link's origin anchor point or table row.
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RetainForm	Gets or sets a value indicating whether a link should maintain its form if the nodes connected with that link are moved around.
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	SegmentCount	Gets or sets the number of segments of this link.
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Shape	Gets or sets the type of link segments and how they are positioned relatively to each other.

	StartPoint	Gets or sets the first control point of a link.
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed over a link.
	TextAlignment	Gets or sets the link's text placement and orientation.
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	TextStyle	Gets or sets the link's text placement and orientation.
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override.
	Visible	DiagramItem.Visible override.
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AddLabel	Overloaded. Adds a label to this link.
	Animate	Animates the item. (Inherited from DiagramItem.)
	Clone	Creates a copy of this link.
	ContainsPoint	Overloaded.
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	Draws the link on the specified IGraphics surface.
S	DrawArrowhead	Draws an arrowhead shape.
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item.
	DrawInRect	
	DrawShadow	Draws the shadow of this link on the specified IGraphics surface.

	GetBounds	Overloaded.
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectiveHeadPen	Returns a Pen object representing the currently effective head pen of the link.
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetLongestHSegment	Returns the index of the longest horizontal segment of this link.
	GetOutlinePoly	DiagramItem.GetOutlinePoly override.
	GetRepaintRect	DiagramItem.GetRepaintRect override.
	GetRotatedBounds	DiagramItem.GetRotatedBounds override.
	GetToolTip	Gets the tooltip that should be displayed for this item at the specified point. (Inherited from DiagramItem.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point.
	HitTestManipulators	Finds the **Manipulator** object located at the specified point. (Inherited from DiagramItem.)
	Intersects	Overloaded.
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point.
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnLoad	DiagramItem.OnLoad override.
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Attaches the link end points to the anchor points defined for the link's origin and destination nodes.
	RemoveLabel	Removes the specified label from this link.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	Route	Routes this link, regardless of whether the link's AutoRoute flag is enabled.
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetOrgAndDest	Sets the origin and destination links of this link.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress.
	UpdateFromPoints	Overloaded. Updates the link's internal state after the link's control points have been changed.
	UpdateIntersections	Aligns the link ends to the contour of the nodes it connects.

	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override.
	AllowModify	DiagramItem.AllowModify override.
	CancelModify	DiagramItem.CancelModify override.
	CompleteCreate	DiagramItem.CompleteCreate override.
	CompleteModify	DiagramItem.CompleteModify override.
	CreateProperties	DiagramItem.CreateProperties override.
	CreateState	DiagramItem.CreateState override.
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the item is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramItem.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	LoadFromJson	Loads the link from a JSON object.
	LoadFromXml	Loads the link content from an XML element.
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	Called when the item is added to a Diagram, either by the user, programmatically or via undo / redo. (Inherited from DiagramItem.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	Called when the Diagram's MeasureUnit changes.
	OnModify	Invoked when the user moves any control point of the link.
	OnRemove	Called when an item is removed from a **Diagram**, either by the user, programmatically or via undo / redo. (Inherited from DiagramItem.)
	OnStyleChanged	Diagramitem.OnStyleChanged override.
	OnTextChanged	DiagramItem.OnTextChanged override.

	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnZoomChanged	Called when the diagram's **ZoomFactor** changes. (Inherited from DiagramItem.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override.
	RestoreState	DiagramItem.RestoreState override.
	SaveProperties	DiagramItem.SaveProperties override.
	SaveState	DiagramItem.SaveState override.
	SaveToJson	Saves the link as a JSON object.
	SaveToXml	Saves the link content into an XML element.
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpanded	Sets whether the item is in expanded or collapsed state. (Inherited from DiagramItem.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	StartCreate	DiagramItem.StartCreate override.
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively.
	UpdateCreate	DiagramItem.UpdateCreate override.
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateModify	DiagramItem.UpdateModify override.

See Also

DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Constructor

Initializes a new instance of the DiagramLink class.

Overload List

Name	Description
DiagramLink ()	Initializes a new instance of the DiagramLink class.
DiagramLink (Diagram)	Initializes a new instance of the **DiagramLink** class with default values supplied form the specified Diagram.
DiagramLink (Diagram, DiagramNode, DiagramNode)	Initializes a new instance of the **DiagramLink** class between the specified node with default values supplied form the specified **Diagram**.
DiagramLink (Diagram, DiagramNode, Point)	Initializes a new instance of the **DiagramLink** class between the specified node and the specified point with default values supplied form the specified **Diagram**.

DiagramLink (Diagram, Point, DiagramNode)	Initializes a new instance of the **DiagramLink** class between the specified point and the specified node with default values supplied form the specified **Diagram**.
DiagramLink (Diagram, Point, Point)	Initializes a new instance of the **DiagramLink** class between the specified points with default values supplied form the specified **Diagram**.
DiagramLink (DiagramLink, DiagramNode, DiagramNode)	Initializes a new instance of the **DiagramLink** class between the specified nodes using the specified diagram link as a prototype.

See Also

DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Constructor ()

Initializes a new instance of the DiagramLink class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramLink Constructor Overload List
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Constructor (Diagram)

Initializes a new instance of the DiagramLink class with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink (
    Diagram parent
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram _
)
```

Parameters

parent
The Diagram from which to obtain default values for the link properties.

See Also

DiagramLink Constructor Overload List
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Constructor (Diagram, DiagramNode, DiagramNode)

Initializes a new instance of the DiagramLink class between the specified node with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink (
    Diagram parent,
    DiagramNode origin,
    DiagramNode destination
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram, _
    origin As DiagramNode, _
    destination As DiagramNode _
)
```

Parameters

parent
The Diagram from which to obtain default values for the link properties.

origin
The origin node of the new link.

destination
The destination node of the new link.

See Also

DiagramLink Constructor Overload List
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Constructor (Diagram, DiagramNode, Point)

Initializes a new instance of the DiagramLink class between the specified node and the specified point with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink (
    Diagram parent,
    DiagramNode origin,
    Point destination
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram, _
    origin As DiagramNode, _
    destination As Point _
)
```

Parameters

parent
The Diagram from which to obtain default values for the link properties.

origin
The origin node of the new link.

destination
The destination point of the new link.

See Also

DiagramLink Constructor Overload List
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Constructor (Diagram, Point, DiagramNode)

Initializes a new instance of the DiagramLink class between the specified point and the specified node with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink (
    Diagram parent,
    Point src,
    DiagramNode dest
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram, _
    src As Point, _
    dest As DiagramNode _
)
```

Parameters

parent
The Diagram from which to obtain default values for the link properties.

src
The origin point of the new link.

dest
The destination node of the new link.

See Also

DiagramLink Constructor Overload List
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Constructor (Diagram, Point, Point)

Initializes a new instance of the DiagramLink class between the specified points with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink (
    Diagram parent,
    Point src,
    Point dest
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram, _
    src As Point, _
    dest As Point _
)
```

Parameters

parent
The Diagram from which to obtain default values for the link properties.

src
The origin point of the new link.

dest
The destination point of the new link.

See Also

DiagramLink Constructor Overload List
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Constructor (DiagramLink, DiagramNode, DiagramNode)

Initializes a new instance of the DiagramLink class between the specified nodes using the specified diagram link as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink (
    DiagramLink prototype,
    DiagramNode src,
    DiagramNode dest
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As DiagramLink, _
    src As DiagramNode, _
    dest As DiagramNode _
)
```

Parameters

prototype
The link to use as a prototype. The properties of the new link are initialized from the property values of this link.

src
The origin node of the new link.

dest
The destination node of the new link.

See Also

DiagramLink Constructor Overload List
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Methods

DiagramLink.AddLabel Method

Adds a label to this link.

Overload List

Name	Description
DiagramLink.AddLabel (LinkLabel)	Adds the specified label to this link.
DiagramLink.AddLabel (String)	Adds a new label to this link.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.AddLabel Method (LinkLabel)

Adds the specified label to this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void AddLabel (
    LinkLabel label
)
```

Visual Basic Copy Code

```
Public Sub AddLabel( _
    label As LinkLabel _
)
```

Parameters

label
A LinkLabel instance.

See Also

AddLabel Method Overload List
DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.AddLabel Method (String)

Adds a new label to this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkLabel AddLabel (
    string text
)
```

Visual Basic Copy Code

```
Public Function AddLabel( _
    text As String _
) As LinkLabel
```

Parameters

text
The label's text.

Return Value

A LinkLabel instance.

See Also

AddLabel Method Overload List
DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.AllowCreate Method

DiagramItem.AllowCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override bool AllowCreate (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Function AllowCreate( _
    current As Point, _
    ist As InteractionState _
) As Boolean
```

Parameters

current
The current mouse pointer position.

ist

An InteractionState object containing more information about the current state of the operation.

Return Value

true to allow the creation of this link; otherwise, **false**.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.AllowModify Method

DiagramItem.AllowModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override bool AllowModify (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Function AllowModify( _
    current As Point, _
    ist As InteractionState _
) As Boolean
```

Parameters

current

The current mouse pointer position.

ist

An InteractionState object containing more information about the current state of the operation.

Return Value

true to allow the modification of this link; otherwise, **false**.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.CancelModify Method

DiagramItem.CancelModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void CancelModify (
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub CancelModify( _
    ist As InteractionState _
)
```

Parameters

ist

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.Clone Method

Creates a copy of this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override DiagramItem Clone (
    bool clipboard
)
```

Visual Basic Copy Code

```
Public Overrides Function Clone( _
    clipboard As Boolean _
) As DiagramItem
```

Parameters

clipboard
true if the copy will be stored in the Windows clipboard; otherwise, **false**.

Return Value

An instance of the DiagramItem class.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.CompleteCreate Method

DiagramItem.CompleteCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void CompleteCreate (
    Point end
)
```

Visual Basic Copy Code

```
Protected Overrides Sub CompleteCreate( _
    end As Point _
)
```

Parameters

end
The mouse pointer position.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.CompleteModify Method

DiagramItem.CompleteModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void CompleteModify (
    Point end,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub CompleteModify( _
    end As Point, _
    ist As InteractionState _
)
```

Parameters

end
The mouse pointer position.

ist
An InteractionState object containing more information about the current state of the operation.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.ContainsPoint Method

Overload List

Name	Description
DiagramLink.ContainsPoint (Point)	DiagramItem.ContainsPoint override.
DiagramLink.ContainsPoint (Point, Double)	DiagramItem.ContainsPoint override.

See Also

DiagramLink.ContainsPoint Method (Point)

DiagramItem.ContainsPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool ContainsPoint (
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function ContainsPoint( _
    point As Point _
) As Boolean
```

Parameters

point
The point to check.

Return Value

true it this link contains the specified point; otherwise, **false**.

See Also

ContainsPoint Method Overload List
DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.ContainsPoint Method (Point, Double)

DiagramItem.ContainsPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool ContainsPoint (
    Point point,
    double threshold
)
```

Visual Basic Copy Code

```
Public Overrides Function ContainsPoint( _
    point As Point, _
    threshold As Double _
) As Boolean
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying the logical coordinates of a diagram point.

threshold
Specifies the size of the area around the item outlines where the point is still considered inside the item.

Return Value

true it this item contains the specified point; otherwise, **false**.

See Also

ContainsPoint Method Overload List
DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.CreateProperties Method

DiagramItem.CreateProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemProperties CreateProperties ()
```

Visual Basic Copy Code

```
Protected Overrides Function CreateProperties () As DiagramItemProperties
```

Return Value

A DiagramItemProperties -derived object containing a snapshot of the item's properties.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.CreateState Method

DiagramItem.CreateState override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemState CreateState ()
```

Visual Basic Copy Code

```
Protected Overrides Function CreateState () As DiagramItemState
```

Return Value

A DiagramLinkState instance.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.Draw Method

Draws the link on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Draw (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub Draw( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
An IGraphics object where the link should be rendered.

options
Provides options for the drawing operation.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.DrawArrowhead Method

Draws an arrowhead shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static void DrawArrowhead (
    IGraphics graphics,
    Pen pen,
    Brush brush,
    Shape shape,
    Point pivot,
    Point position,
    Point segmentStart,
    double size,
    bool shadow
)
```

Visual Basic Copy Code

```
Public Shared Sub DrawArrowhead( _
    graphics As IGraphics, _
    pen As Pen, _
    brush As Brush, _
    shape As Shape, _
    pivot As Point, _
    position As Point, _
    segmentStart As Point, _
```

```
    size As Double, _
    shadow As Boolean _
)
```

Parameters

graphics
The graphics surface to draw the shape on.

pen
A Pen instance used to draw the shape outline.

brush
A Brush instance used to fill the shape.

shape
A Shape instance that defines the arrowhead geometry.

pivot
A **Microsoft.Maui.Graphics.Point** specified as percentage that corresponds to the arrowhead tip position within the **Shape** geometry.

position
A point on the drawing surface where the arrowhead tip should be located.

segmentStart
A point specifying the arrowhead direction.

size
A double value that specifies the arrowhead size.

shadow
A bool value that specifies if drawing shadow, which causes the passed brush and pen to be used instead of those provided by the shape.

Remarks

This method lets you draw any Shape object at arbitrary position and angle, and is useful for rendering arrowhead shapes when custom-drawing links.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
CustomDraw Property
Diagram.DrawLink Event

DiagramLink.DrawHandles Method

DiagramItem.DrawHandles override. Renders the adjustment handles of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawHandles (
    IGraphics graphics,
    HandlesVisualStyle style,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawHandles( _
    graphics As IGraphics, _
    style As HandlesVisualStyle, _
    options As RenderOptions _
)
```

Parameters

graphics

The IGraphics surface on which the handles should be drawn.

style

A HandlesVisualStyle instance containing brush and pen objects that should be used when drawing handles.

options

Provides options for the drawing operation.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
HitTestHandle Method
HandlesStyle Property

DiagramLink.DrawInRect Method

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawInRect (
    IGraphics graphics,
    RenderOptions options,
    Rect target
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawInRect( _
    graphics As IGraphics, _
    options As RenderOptions, _
    target As Rect _
)
```

Parameters

graphics

options

target

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.DrawShadow Method

Draws the shadow of this link on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawShadow (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawShadow( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
An IGraphics object where the shadow should be rendered.

options
A RenderOptions object specifying options for the drawing operation.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.GetBounds Method

Overload List

Name	Description
DiagramLink.GetBounds ()	DiagramItem.GetBounds override.
DiagramLink.GetBounds (Boolean)	Gets the bounding rectangle of this link.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.GetBounds Method ()

DiagramItem.GetBounds override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Rect GetBounds ()
```

Visual Basic Copy Code

```
Public Overrides Function GetBounds () As Rect
```

Return Value

The rectangle that bounds this link's control points.

See Also

GetBounds Method Overload List
DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.GetBounds Method (Boolean)

Gets the bounding rectangle of this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect GetBounds (
    bool withStrokeThickness
)
```

Visual Basic Copy Code

```
Public Function GetBounds( _
    withStrokeThickness As Boolean _
) As Rect
```

Parameters

withStrokeThickness
true to consider link's pen width when calculating the bounding rectangle, or **false** otherwise.

Return Value

A **Microsoft.Maui.Graphics.Rect** instance containing the coordinates of link's bounding rectangle.

See Also

GetBounds Method Overload List
DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.GetEffectiveHeadPen Method

Returns a Pen object representing the currently effective head pen of the link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen GetEffectiveHeadPen (
    IGraphics graphics
)
```

Visual Basic Copy Code

```
Public Function GetEffectiveHeadPen( _
    graphics As IGraphics _
) As Pen
```

Parameters

graphics

Return Value

Remarks

This method returns the link's HeadPen if the link's HeadPen is not null, or a pen with the HeadStroke and HeadStrokeThickness properties of the link's style.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.GetLongestHSegment Method

Returns the index of the longest horizontal segment of this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int GetLongestHSegment ()
```

Visual Basic Copy Code

```
Public Function GetLongestHSegment () As Integer
```

Return Value

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.GetOutlinePoly Method

DiagramItem.GetOutlinePoly override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override PointCollection GetOutlinePoly (
    bool rendering
)
```

Visual Basic Copy Code

```
Public Overrides Function GetOutlinePoly( _
    rendering As Boolean _
) As PointCollection
```

Parameters

rendering

Return Value

See Also

DiagramLink.GetRepaintRect Method

DiagramItem.GetRepaintRect override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Rect GetRepaintRect (
    bool includeConnected
)
```

Visual Basic Copy Code

```
Public Overrides Function GetRepaintRect( _
    includeConnected As Boolean _
) As Rect
```

Parameters

includeConnected
true if related items should be considered too; otherwise, **false**.

Return Value

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.GetRotatedBounds Method

DiagramItem.GetRotatedBounds override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Rect GetRotatedBounds ()
```

Visual Basic Copy Code

```
Public Overrides Function GetRotatedBounds () As Rect
```

Return Value

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.HitTestHandle Method

DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override AdjustmentHandle HitTestHandle (
    Point point,
    double distance
)
```

Visual Basic Copy Code

```
Public Overrides Function HitTestHandle( _
    point As Point, _
    distance As Double _
) As AdjustmentHandle
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** value specifying where to look for a selection handle.

distance

Return Value

A LinkAdjustmentHandle instance if there is a control point at the specified location, otherwise **null**.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
DrawHandles Method
HandlesStyle Property

DiagramLink.Intersects Method

Overload List

Name	Description
DiagramLink.Intersects (DiagramLink)	Checks whether this link intersects with the specified link.
DiagramLink.Intersects (DiagramNode)	Checks whether the link contour intersects with the outline of the specified node.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.Intersects Method (DiagramLink)

Checks whether this link intersects with the specified link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Intersects (
    DiagramLink link
)
```

Visual Basic Copy Code

```
Public Function Intersects( _
    link As DiagramLink _
) As Boolean
```

Parameters

link

Return Value

See Also

Intersects Method Overload List
DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.Intersects Method (DiagramNode)

Checks whether the link contour intersects with the outline of the specified node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Intersects (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Function Intersects( _
    node As DiagramNode _
) As Boolean
```

Parameters

node
A DiagramNode object for which to determine whether the link intersects it.

Return Value

true if the link intersects the node; otherwise, **false**.

See Also

Intersects Method Overload List
DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.LabelFromPoint Method

Returns the label whose layout rectangle contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override ItemLabel LabelFromPoint (
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function LabelFromPoint( _
    point As Point _
) As ItemLabel
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** instance.

Return Value

The LinkLabel that contains the specified point.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.LoadFromJson Method

Loads the link from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the link's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.LoadFromXml Method

Loads the link content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** containing the item's data.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.OnLoad Method

DiagramItem.OnLoad override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void OnLoad ()
```

Visual Basic Copy Code

```
Public Overrides Sub OnLoad ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.OnMeasureUnitChanged Method

Called when the Diagram's MeasureUnit changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnMeasureUnitChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnMeasureUnitChanged ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.OnModify Method

Invoked when the user moves any control point of the link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnModify (
    Point mousePosition,
    int adjustmentHandle
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnModify( _
    mousePosition As Point, _
    adjustmentHandle As Integer _
)
```

Parameters

mousePosition
Specifies the mouse position.

adjustmentHandle
Specifies the adjustment handle used to modify the link.

Remarks

Implement this method in a custom link class if you need to detect modifications done by the user without handling events of the Diagram class.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.OnStyleChanged Method

Diagramitem.OnStyleChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnStyleChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnStyleChanged ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.OnTextChanged Method

DiagramItem.OnTextChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnTextChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnTextChanged ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.ReassignAnchorPoints Method

Attaches the link end points to the anchor points defined for the link's origin and destination nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ReassignAnchorPoints ()
```

Visual Basic Copy Code

```
Public Sub ReassignAnchorPoints ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.RemoveLabel Method

Removes the specified label from this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RemoveLabel (
    LinkLabel label
)
```

Visual Basic Copy Code

```
Public Sub RemoveLabel( _
    label As LinkLabel _
)
```

Parameters

label
A LinkLabel instance.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.RestoreProperties Method

DiagramItem.RestoreProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties -derived object containing a snapshot of the item's properties.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.RestoreState Method

DiagramItem.RestoreState override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreState (
    DiagramItemState state
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreState( _
    state As DiagramItemState _
)
```

Parameters

state
A DiagramLinkState instance.

Remarks

This method is called when a link should be restored to the state it had before user interaction.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.Route Method

Routes this link, regardless of whether the link's AutoRoute flag is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Route ()
```

Visual Basic Copy Code

```
Public Function Route () As Boolean
```

Return Value

true is a route has been successfully found; otherwise, **false**.

Remarks

Routing may fail if there is a large background node whose **Obstacle** property is enabled, or if the value of RoutingOptions.GridSize is too big for the average distance between adjacent nodes in the diagram.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.SaveProperties Method

DiagramItem.SaveProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties -derived object containing a snapshot of the item's properties.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.SaveState Method

DiagramItem.SaveState override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveState (
    DiagramItemState state
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveState( _
    state As DiagramItemState _
)
```

Parameters

state
A DiagramLinkState instance.

Remarks

This method is called to save undo information when the users starts modifying a link.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.SaveToJson Method

Saves the link as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject instance to contain the serialized data.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.SaveToXml Method

Saves the link content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** the link's data should be stored into.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.SetOrgAndDest Method

Sets the origin and destination links of this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetOrgAndDest (
    ConnectionPoint orgConnectionPoint,
    ConnectionPoint trgConnectionPoint
)
```

Visual Basic Copy Code

```
Public Sub SetOrgAndDest( _
    orgConnectionPoint As ConnectionPoint, _
    trgConnectionPoint As ConnectionPoint _
)
```

Parameters

orgConnectionPoint

trgConnectionPoint

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.StartCreate Method

DiagramItem.StartCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void StartCreate (
    Point org
)
```

Visual Basic Copy Code

```
Protected Overrides Sub StartCreate( _
    org As Point _
)
```

Parameters

org
Position of the mouse pointer.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.StartModify Method

DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void StartModify (
    Point org,
    AdjustmentHandle handle,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub StartModify( _
    org As Point, _
    handle As AdjustmentHandle, _
    ist As InteractionState _
)
```

Parameters

org
Position of the mouse pointer.

handle
The adjustment handle used to modify this link.

ist
An InteractionState object containing more information about the current state of the operation.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.UpdateAnimation Method

Invoked by running animations to update an item according to current animation progress.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void UpdateAnimation (
    Animation animation,
    double animationDelta
)
```

Visual Basic Copy Code

```
Public Overridable Sub UpdateAnimation( _
    animation As Animation, _
    animationDelta As Double _
)
```

Parameters

animation
The Animation object used to animate this link.

animationDelta

A value representing the current animation progress

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.UpdateCreate Method

DiagramItem.UpdateCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void UpdateCreate (
    Point current
)
```

Visual Basic Copy Code

```
Protected Overrides Sub UpdateCreate( _
    current As Point _
)
```

Parameters

current
Position of the mouse pointer.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.UpdateFromPoints Method

Updates the link's internal state after the link's control points have been changed.

Overload List

Name	Description
DiagramLink.UpdateFromPoints ()	Updates the link's internal state after the link's control points have been changed.
DiagramLink.UpdateFromPoints (Boolean)	Updates the link's internal state after the link's control points have been changed.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.UpdateFromPoints Method ()

Updates the link's internal state after the link's control points have been changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void UpdateFromPoints ()
```

Visual Basic Copy Code

```
Public Sub UpdateFromPoints ()
```

See Also

UpdateFromPoints Method Overload List
DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.UpdateFromPoints Method (Boolean)

Updates the link's internal state after the link's control points have been changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void UpdateFromPoints (
    bool updateGroups
)
```

Visual Basic Copy Code

```
Public Sub UpdateFromPoints( _
    updateGroups As Boolean _
)
```

Parameters

updateGroups
true to update the positions of nodes attached to the link; otherwise, **false**.

See Also

UpdateFromPoints Method Overload List
DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.UpdateIntersections Method

Aligns the link ends to the contour of the nodes it connects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void UpdateIntersections ()
```

Visual Basic Copy Code

```
Public Sub UpdateIntersections ()
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.UpdateModify Method

DiagramItem.UpdateModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void UpdateModify (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub UpdateModify( _
    current As Point, _
    ist As InteractionState _
)
```

Parameters

current
Position of the mouse pointer.

ist
An InteractionState object containing more information about the current state of the operation.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink Properties

DiagramLink.AllowMoveEnd Property

Gets or sets a value indicating whether the user is allowed to move the end point of a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowMoveEnd { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowMoveEnd As Boolean
```

Property Value

true to enable users to move the end point of the link; otherwise, **false**. Initialized with the value of LinkEndsMovable.

Remarks

Set the property to **false** to prevent the link destination to be changed interactively. Alternatively, you might handle the LinkModifying event to implement selective validation of attaching a link to different diagram nodes.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
AllowMoveStart Property
Diagram.LinkEndsMovable Property

DiagramLink.AllowMoveStart Property

Gets or sets a value indicating whether the user is allowed to move the first control point of a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowMoveStart { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowMoveStart As Boolean
```

Property Value

true to enable users to move the start point of the link; otherwise **false**. Initialized with the value of LinkEndsMovable.

Remarks

Set the property to **false** to prevent the link origin to be changed interactively. Alternatively, you might handle the LinkModifying event to implement selective validation of attaching a link to different diagram nodes.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
AllowMoveEnd Property
Diagram.LinkEndsMovable Property

DiagramLink.AutoRoute Property

Gets or sets a value indicating whether a link should avoid nodes by going the shortest path from its origin to its destination without crossing any other nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoRoute { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoRoute As Boolean
```

Property Value

true to route this link; otherwise, **false**. Initialized with the value of RouteLinks.

Remarks

If **AutoRoute** is enabled, a link automatically finds the shortest possible path between its origin and destination that doesn't intersect **Obstacle** nodes. The number and orientation of the link's segments are updated when the nodes it links are moved around. The positions of end points of a routed link are not changed, unless the link is also set to Dynamic.

It is best to use **AutoRoute** with links of the Cascading style. Note that automatic routing overrides the RetainForm behavior.

Performance note

Links whose **AutoRoute** property is enabled are re-routed if they intersect with newly created nodes. Thus when constructing a large diagram programmatically, each link might be re-routed many times, once for each newly added node whose location arguments collide with link's bounding rectangle. That might slow down diagram creation a lot. To avoid it, first create all nodes, and afterwards the link; that requires a single calculation for each link's route.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
RoutingOptions Class
Dynamic Property
SegmentCount Property
LinkRouter Property

DiagramLink.AutoSnapToNode Property

Specifies whether a link end point should automatically snap to the nearest diagram node while the user moves the point's adjustment handle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoSnapToNode { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoSnapToNode As Boolean
```

Property Value

true to snap links to nearest node; otherwise, **false**. Initialized with the value of AutoSnapLinks.

Remarks

Only nodes at a distance smaller than AutoSnapDistance are considered. A link does not snap to its current origin or destination nodes if that would create a self-loop. If there is an **AnchorPattern** assigned to a node, the link snaps to an anchor point instead of a border point.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
Dynamic Property
RetainForm Property

DiagramLink.BaseBrush Property

Gets or sets a brush used to fill the arrowhead drawn at the link base.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush BaseBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property BaseBrush As Brush
```

Property Value

An instance of a Brush -derived class. The default value is **null** (**Nothing** in Visual Basic).

Remarks

If this property is not set, the arrowhead is filled using the link's Brush.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
BaseShape Property

DiagramLink.BaseShape Property

Gets or sets what shape to display at the beginning of a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape BaseShape { get; set; }
```

Visual Basic Copy Code

```
Public Property BaseShape As Shape
```

Property Value

A Shape instance that represents an arrowhead shape. Initialized with the value of LinkBaseShape.

Remarks

The property specifies the shape that is displayed at the beginning of a link, that is, the point where it is connected with its Origin node. This shape is rotated simultaneously with the first segment of the link so that it follows the overall link direction.

The ArrowHeads class exposes several predefined shapes as static properties.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
HeadShape Property
IntermediateShape Property
BaseBrush Property
BaseShapeSize Property

DiagramLink.BaseShapeSize Property

Gets or sets the size of the BaseShape shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double BaseShapeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property BaseShapeSize As Double
```

Property Value

A **double** value that specifies the shape size. Initialized with the value of LinkBaseShapeSize.

Remarks

The property specifies how big is the shape that is displayed at the origin of a link.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
BaseShape Property

DiagramLink.Bounds Property

Gets the size of the rectangle that bounds this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect Bounds { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Bounds As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** that represents the bounding rectangle.

Remarks

Use the property to get the smallest rectangle that bounds all link points.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.CascadeOrientation Property

Gets or sets the orientation of the first segment of a cascading link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Orientation CascadeOrientation { get; set; }
```

Visual Basic Copy Code

```
Public Property CascadeOrientation As Orientation
```

Property Value

A member of the Orientation enumeration.

Remarks

Use this property to specify the orientation of the first segment of a link whose Shape is set to Cascading. See Orientation for more details. The orientation of the first segment defines the orientation of all other segments in the cascade, because each next segment changes its orientation alternately to the previous one.

If the property is set to Auto, users may change the orientation of a cascading link interactively. That can be done by dragging the control point next to the first or last one to its opposite corner in the rectangle defined by its two adjacent segments. Depending on the value of AllowSplitLinks, the link orientation is changed by either inserting a new segment and changing only the orientation of the first or last segment (**AllowSplitLinks** is **true**), or by changing the orientation of all segments of the link (**AllowSplitLinks** is **false**).

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
Diagram.LinkCascadeOrientation Property
Shape Property

DiagramLink.ControlPoints Property

Gets a collection of the control points of this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PointCollection ControlPoints { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ControlPoints As PointCollection
```

Property Value

A collection of control points that specify the link's shape.

Remarks

You can specify new positions of a link's points through that property. The number of points in the collection depends on the Shape and SegmentCount of the link. Links made of Bézier-splines has four control points for each segment. Links of the other styles have two control points per segment. Adjacent segments share a common control point. Check the Count property of the collection to get the number of points. After setting new positions of control points you must call the UpdateFromPoints method.

Example

The following example demonstrates how to programmatically setup the control points of a link:

C# Copy Code

```
DiagramLink link;
Point cpt;

link = Diagram.Links[0];

// Move the second and the third points
cpt = link.ControlPoints[1];
link.ControlPoints[1] = new Point(cpt.X + 5, cpt.Y);
link.ControlPoints[2] = new Point(22.2, 33.3);

// Update state and redraw
link.UpdateFromPoints();
```

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
UpdateFromPoints Method
StartPoint Property
EndPoint Property

DiagramLink.CrossingRadius Property

Gets or sets the radius length of decorations displayed at intersection points of this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double CrossingRadius { get; set; }
```

Visual Basic Copy Code

```
Public Property CrossingRadius As Double
```

Property Value

A **double** value specifying the radius of arcs or cuts displayed at the intersection points of this link. The default is 0.

Remarks

If this property is set to 0, the control uses the value of the Diagram.CrossingRadius property.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
DrawCrossings Property

DiagramLink.CustomDraw Property

Gets or sets the type of custom drawing performed on this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CustomDraw CustomDraw { get; set; }
```

Visual Basic Copy Code

```
Public Property CustomDraw As CustomDraw
```

Property Value

A member of the CustomDraw enumeration. Initialized with the value of LinkCustomDraw.

Remarks

This property allows invoking custom painting code for a link. That code must execute in response to the DrawLink event. There are several modes of custom painting, allowing the custom code to paint the link entirely, or do that in addition to what MvcDiagram paints by default. The following CustomDraw members can be assigned to this property to specify the custom painting mode:

Member name	Description
None	Do not perform custom drawing, let all painting be done by MvcDiagram.
Additional	The custom painting code executes after the link segments and arrowhead shapes are painted by MvcDiagram, but before text is rendered.
Full	The component does not do any standard drawing, the entire painting of the link is done by the application.
ShadowOnly	Only the shadow of the link is painted by custom code, the link itself is drawn by the component.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.Destination Property

Gets or sets the destination node of this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Destination { get; set; }
```

Visual Basic Copy Code

```
Public Property Destination As DiagramNode
```

Property Value

A reference to the destination DiagramNode of the link.

Remarks

The link's destination can be changed programmatically with this property. If any of the connected nodes has an **AnchorPattern** defined, links ends are placed at the nearest anchor points of the two nodes. If AutoRoute is enabled, the link is routed when changing its origin or destination.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
Origin Property

DestinationAnchor Property
Query Method

DiagramLink.DestinationAnchor Property

Gets or sets the index of the destination anchor point within the AnchorPattern that contains it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int DestinationAnchor { get; set; }
```

Visual Basic Copy Code

```
Public Property DestinationAnchor As Integer
```

Property Value

An integer index of an anchor point within the AnchorPattern containing it.

Remarks

Specifies the index of an anchor point contained within the Destination anchor pattern. The link is docked and stay connected to that point if the node is moved or resized.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
OriginAnchor Property
Diagram.DestinationAnchorChanged Event

DiagramLink.DestinationConnection Property

Gets or sets the ConnectionPoint object representing the link connection to its destination node. This allows connecting the link to a constituent part of a node, such as a TableNode row or a TreeViewItem.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ConnectionPoint DestinationConnection { get; set; }
```

Visual Basic Copy Code

```
Public Property DestinationConnection As ConnectionPoint
```

Property Value

An instance of ConnectionPoint or a derived class.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
Destination Property

DiagramLink.DestinationIndex Property

Gets or sets the index of link's destination anchor point or table row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int DestinationIndex { get; set; }
```

Visual Basic Copy Code

```
Public Property DestinationIndex As Integer
```

Property Value

An integer value specifying the index of table's row or the index of an anchor point within its containing AnchorPattern.

Remarks

If a link's destination is a shape node, this property lets you get or set the index of an anchor point at which the link docks to the node. If there is no AnchorPattern defined for the destination node, the property returns -1. If the link is connected to a table, the property returns or changes the related row of that table.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
OriginIndex Property

DiagramLink.DrawCrossings Property

Gets or sets a value indicating whether the points where this link crosses other links should be drawn as arcs or dashes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool DrawCrossings { get; set; }
```

Visual Basic Copy Code

```
Public Property DrawCrossings As Boolean
```

Property Value

true if crossing points should be drawn as arcs or dashes; otherwise, **false**. The default is **true**.

Remarks

The LinkCrossings property of the Diagram class specifies what shapes are drawn at link crossing points. Individual links cannot display shapes different from the ones drawn for other links, but the crossing points shapes can be shown or hidden for a link through this property.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
CrossingRadius Property

DiagramLink.Dynamic Property

Gets or sets a value indicating whether a link automatically adjusts its end points positions in order to keep pointing to the centers of nodes that it connects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Dynamic { get; set; }
```

Visual Basic Copy Code

```
Public Property Dynamic As Boolean
```

Property Value

true to make the link automatically adjust its end points; otherwise, **false**. Initialized with the value of DynamicLinks.

Remarks

Use the property to specify whether a link automatically repositions its end segments, so they keep pointing to the centers of nodes connected by that link. If enabled, moving a connected node recalculates the position of link's end point, placing it at appropriate location of the node outline. If the node has anchor points assigned to it, the end point is attached to one of them. End points are realigned also when moving the points next to them.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
RetainForm Property

DiagramLink.EffectiveHeadPen Property

Gets the effective head pen of the link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen EffectiveHeadPen { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectiveHeadPen As Pen
```

Property Value

An instance of the MindFusion.Drawing.Pen class.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.EndPoint Property

Gets or sets the last control point of a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point EndPoint { get; set; }
```

Visual Basic Copy Code

```
Public Property EndPoint As Point
```

Property Value

A **Microsoft.Maui.Graphics.Point** object.

Remarks

This property is a shortcut for getting or setting the last element of ControlPoints.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
StartPoint Property

DiagramLink.Font Property

DiagramItem.Font override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Font Font { get; set; }
```

Visual Basic Copy Code

```
Public Overrides Property Font As Font
```

Property Value

The Font to use when drawing the link's text.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.HandlesStyle Property

Gets or sets the appearance of the link's selection handles.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HandlesStyle HandlesStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property HandlesStyle As HandlesStyle
```

Property Value

A member of the HandlesStyle enumeration. Initialized with the value of LinkHandlesStyle.

Remarks

Specifies how link's selection status is indicated visually. You can choose between the following styles:

Member name	Description
SquareHandles	The selection is indicated by drawing square handles at the position of each control point.
DashFrame	The selection is indicated by drawing dashed line segments along the link's segments. In addition, there are square selection handles drawn at the location of each control point of the link.
Custom	MvcDiagram raises the DrawAdjustmentHandles events to let your application perform custom drawing of link's selection handles.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
DrawHandles Method
HitTestHandle Method

DiagramLink.HeadBrush Property

Gets or sets a brush used to fill the arrowhead drawn at the link's destination.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush HeadBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property HeadBrush As Brush
```

Property Value

An instance of a Brush -derived class. The default value is **null** (**Nothing** in Visual Basic).

Remarks

If this property is not set, the arrowhead is filled using the link's Brush.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
HeadShape Property

DiagramLink.HeadPen Property

Gets or sets the pen to use when painting arrowheads lines.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen HeadPen { get; set; }
```

Visual Basic Copy Code

```
Public Property HeadPen As Pen
```

Property Value

An instance of the MindFusion.Drawing.Pen class. Initialized with a clone of LinkPen.

Remarks

This property refers to a pen used to paint arrowhead lines. To change how segment lines are painted, use the Pen property.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
Brush and Pen Classes
Brush Property

DiagramLink.HeadShape Property

Gets or sets the shape of this link's arrowhead.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape HeadShape { get; set; }
```

Visual Basic Copy Code

```
Public Property HeadShape As Shape
```

Property Value

A Shape instance that represents an arrowhead shape. Initialized with the value of LinkHeadShape.

Remarks

The property specifies what shape is displayed as an arrowhead. This shape is rotated simultaneously with the last segment of the link so that it follows the overall link direction.

The ArrowHeads class exposes several predefined shapes as static properties.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
BaseShape Property
IntermediateShape Property
HeadBrush Property
HeadShapeSize Property

DiagramLink.HeadShapeSize Property

Gets or sets the size of the arrowhead shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double HeadShapeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property HeadShapeSize As Double
```

Property Value

A **double** value that measures the size. Initialized with the value of LinkHeadShapeSize.

Remarks

The property specifies how big the shape of the arrowhead is.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
HeadShape Property

DiagramLink.IntermediateBrush Property

Gets or sets a brush used to fill the arrowheads drawn at the middle of link segments.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush IntermediateBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property IntermediateBrush As Brush
```

Property Value

An instance of a Brush -derived class. The default value is **null** (**Nothing** in Visual Basic).

Remarks

If this property is not set, the arrowheads are filled using the link's Brush.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
IntermediateShape Property

DiagramLink.IntermediateShape Property

Gets or sets the shape displayed at the middle of each segment of the link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape IntermediateShape { get; set; }
```

Visual Basic Copy Code

```
Public Property IntermediateShape As Shape
```

Property Value

A Shape instance that represents an arrowhead shape. Initialized with the value of LinkIntermediateShape.

Remarks

The property specifies what shape to display at the middle of each segment of a link. This could help users to see more easily the direction of long links. The ArrowHeads class exposes several predefined shapes as static properties.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
HeadShape Property
BaseShape Property
IntermediateBrush Property
IntermediateShapeSize Property

DiagramLink.IntermediateShapeSize Property

Gets or sets the size of the IntermediateShape shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double IntermediateShapeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property IntermediateShapeSize As Double
```

Property Value

A **double** value that measures the size. Initialized with the value of LinkIntermediateShapeSize.

Remarks

The property specifies how big are the shapes that are displayed in the middle of link segments.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
IntermediateShape Property

DiagramLink.IsConnected Property

Gets a value indicating whether the link is connected to nodes at both its ends.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsConnected { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property IsConnected As Boolean
```

Property Value

true to indicate that both ends of the link are connected to nodes; otherwise, **false**.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.Labels Property

Returns the list of all labels displayed by this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<LinkLabel> Labels { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Labels As List(Of LinkLabel)
```

Property Value

A list with all link labels.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.Length Property

Gets the length of the link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Length { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Length As Double
```

Property Value

A **double** value that represents the total length of the link, sum of the lengths of all its segments.

Remarks

If the link's Shape is set to Bezier, an approximate length is calculated. In that case, the approximate value is almost always slightly smaller than the real length of Bézier segments.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
Shape Property
SegmentCount Property

DiagramLink.Origin Property

Gets or sets the origin node of the link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Origin { get; set; }
```

Visual Basic Copy Code

```
Public Property Origin As DiagramNode
```

Property Value

A reference to the origin DiagramNode of the link.

Remarks

The link's origin can be changed programmatically with this property. If any of the connected nodes has an AnchorPattern defined, links ends are placed at the nearest anchor points of the two nodes. If AutoRoute is enabled, the link is routed when changing its origin or destination.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
Destination Property
OriginAnchor Property
Query Method

DiagramLink.OriginAnchor Property

Gets or sets the index of the origin anchor point within the AnchorPattern that contains it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int OriginAnchor { get; set; }
```

Visual Basic Copy Code

```
Public Property OriginAnchor As Integer
```

Property Value

An integer index of an anchor point within its the AnchorPattern containing it.

Remarks

Specifies the index of an anchor point contained within the Origin anchor pattern. The link is docked and stay connected to that point if the node is moved or resized.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
DestinationAnchor Property
Diagram.OriginAnchorChanged Event

DiagramLink.OriginConnection Property

Gets or sets the ConnectionPoint object representing the link connection to its origin node. This allows connecting the link to a constituent part of a node, such as a TableNode row or a TreeViewItem.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ConnectionPoint OriginConnection { get; set; }
```

Visual Basic Copy Code

```
Public Property OriginConnection As ConnectionPoint
```

Property Value

An instance of ConnectionPoint or derived class.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
Origin Property

DiagramLink.OriginIndex Property

Gets or sets the index of link's origin anchor point or table row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int OriginIndex { get; set; }
```

Visual Basic Copy Code

```
Public Property OriginIndex As Integer
```

Property Value

An integer value specifying the index of table's row or index of an anchor point within its containing AnchorPattern.

Remarks

If a link's origin is a shape node, this property lets you get or set the index of an anchor point at which the link docks to the node. If there is no AnchorPattern defined for the origin node, the property returns -1. If the link is connected to a table, the property returns or changes the related row of that table.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
DestinationIndex Property

DiagramLink.RetainForm Property

Gets or sets a value indicating whether a link should maintain its form if the nodes connected with that link are moved around.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RetainForm { get; set; }
```

Visual Basic Copy Code

```
Public Property RetainForm As Boolean
```

Property Value

true to retain the form of this link; otherwise, **false**. Initialized with the value of LinksRetainForm.

Remarks

RetainForm is useful for links that have more than two control points. If the nodes that a link connects are moved around, usually only the segments at the ends of that link are updated, so that they stay connected to their respective nodes. In that case all intermediate control points stay where they are. However, if **RetainForm** is enabled, all control points will move simultaneously in a way they preserve their relative position to each other, and thus preserving the overall form of the link.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
AutoRoute Property
Dynamic Property

DiagramLink.SegmentCount Property

Gets or sets the number of segments of this link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int SegmentCount { get; set; }
```

Visual Basic Copy Code

```
Public Property SegmentCount As Integer
```

Property Value

An integer value specifying how many segments the link consists of. Initialized with the value of LinkSegments.

Remarks

Use this property to make a link more indented when necessary. Its value, together with Shape, specifies the number of points in the link's ControlPoints collection. You can let users insert new link segments interactively by setting AllowSplitLinks to **true**.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
ControlPoints Property
Length Property

DiagramLink.Shape Property

Gets or sets the type of link segments and how they are positioned relatively to each other.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkShape Shape { get; set; }
```

Visual Basic Copy Code

```
Public Property Shape As LinkShape
```

Property Value

A member of the LinkShape enumeration. Initialized with the value of LinkShape.

Remarks

This property specifies whether the segments of a link are straight lines or curves. Its value together with SegmentCount also specifies the number of ControlPoints. Each Bezier segment is defined by four control points, where the second and third ones do not lie on the segment but specify its curvature. All other **LinkShape** types are defined by two control points per segment. Adjacent segments share a control point, and when moved it will modify both of them.

Cascading links consist of straight lines whose orientation is constrained to strictly horizontal or vertical, where adjacent segments are orthogonal to each other. The orientation of the first segment of a cascading link depends on the value of CascadeOrientation. If that property is set to Auto, users may change the cascade orientation interactively.

Each control point of a **Spline** or **Polyline** link can be moved freely, independently from each other, and the link will pass through all control points drawn respectively as an interpolating spline curve or as a polygonal chain.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
ControlPoints Property
AutoRoute Property

DiagramLink.StartPoint Property

Gets or sets the first control point of a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point StartPoint { get; set; }
```

Visual Basic Copy Code

```
Public Property StartPoint As Point
```

Property Value

A **Microsoft.Maui.Graphics.Point** object.

Remarks

This property is a shortcut for getting or setting ControlPoints[0].

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
EndPoint Property

DiagramLink.Text Property

Gets or sets the text displayed over a link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Text { get; set; }
```

Visual Basic Copy Code

```
Public Property Text As String
```

Property Value

A string containing the text. Initialized with the value of LinkText.

Remarks

Gets or sets the text that should be drawn over a link. Use this to provide to end-users a short description of the relation's purpose. The text appearance can be customized through the TextBrush and TextStyle properties.

Version 6 of the control adds the capability to display multiple labels per link via the LinkLabel class. Labels can automatically avoid overlapping nodes or other labels when their AutoArrange property is set.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
ToolTip Property
AddLabel Method

DiagramLink.TextAlignment Property

Gets or sets the link's text placement and orientation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringAlignment TextAlignment { get; set; }
```

Visual Basic Copy Code

```
Public Property TextAlignment As StringAlignment
```

Property Value

A member of the **StringAlignment** enumeration. The default value is **Center**.

Remarks

Currently this property is considered only when TextStyle is set to OverLongestSegment.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.TextStyle Property

Gets or sets the link's text placement and orientation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkTextStyle TextStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property TextStyle As LinkTextStyle
```

Property Value

A member of the LinkTextStyle enumeration. Initialized with the value of LinkTextStyle.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace
Text Property
TextColor Property

DiagramLink.TopLevel Property

DiagramItem.TopLevel override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool TopLevel { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property TopLevel As Boolean
```

Property Value

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLink.Visible Property

DiagramItem.Visible override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool Visible { get; set; }
```

Visual Basic Copy Code

```
Public Overrides Property Visible As Boolean
```

Property Value

true if the item is visible; otherwise, **false**.

See Also

DiagramLink Members
DiagramLink Class
MindFusion.Diagramming Namespace

DiagramLinkCollection Class

Represents a strongly typed collection of DiagramLink objects.

Syntax

C# Copy Code

```
public class DiagramLinkCollection : ItemCollectionBase<DiagramLink>
```

Visual Basic Copy Code

```
Public Class DiagramLinkCollection
    Inherits ItemCollectionBase(Of DiagramLink)
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.CollectionBase(DiagramLink)
    MindFusion.Diagramming.ItemCollectionBase(DiagramLink)
      MindFusion.Diagramming.DiagramLinkCollection

See Also

DiagramLinkCollection Members
MindFusion.Diagramming Namespace

DiagramLinkCollection Members

Represents a strongly typed collection of DiagramLink objects.

The following tables list the members exposed by the DiagramLinkCollection type.

Public Constructors

	Name	Description
	DiagramLinkCollection	Initializes a new instance of the **DiagramLinkCollection** class.

Protected Fields

	Name	Description
	InnerList	Gets a **List<T>** containing the list of elements in the collection instance. (Inherited from CollectionBase.)
	itemEventArgs	A ItemEventArgs instance used for passing arguments to the event handlers for the events of this collection. Used to avoid creating many small objects that would burden the garbage collector. (Inherited from ItemCollectionBase.)
	notifyParent	If enabled, the collection raises events to let their containing objects handle the addition or removal of items. (Inherited from ItemCollectionBase.)

Public Properties

	Name	Description
	Count	Gets the number of elements actually contained in the collection. (Inherited from CollectionBase.)
	IsReadOnly	Gets a value indicating whether the collection is read-only. (Inherited from CollectionBase.)
	Item	Gets or sets the element at the specified index. (Inherited from CollectionBase.)

Public Methods

	Name	Description
	Add	Adds the specified diagram item to the collection. (Inherited from ItemCollectionBase.)
	Clear	Removes all diagram items from the collection. (Inherited from ItemCollectionBase.)
	Clone	Creates a shallow copy of this collection.
	Contains	Determines whether an element is in the collection. (Inherited from CollectionBase.)
	CopyTo	Copies the entire collection to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from CollectionBase.)
	GetAt	Gets the diagram item at the specified position. (Inherited from ItemCollectionBase.)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from CollectionBase.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire collection. (Inherited from CollectionBase.)
	Insert	Inserts the specified diagram item into the collection at the specified position. (Inherited from ItemCollectionBase.)
	Remove	Removes the diagram item from the collection. (Inherited from ItemCollectionBase.)

	RemoveAt	Removes the diagram item from the collection. (Inherited from ItemCollectionBase.)
	Sort	Overloaded. (Inherited from CollectionBase.)

Protected Methods

	Name	Description
	RaiseAdding	Raises the **Adding** event. (Inherited from ItemCollectionBase.)
	RaiseInserting	Raises the **Inserting** event. (Inherited from ItemCollectionBase.)
	RaiseRemoving	Raises the **Removing** event. (Inherited from ItemCollectionBase.)

See Also

DiagramLinkCollection Class
MindFusion.Diagramming Namespace

DiagramLinkCollection Constructor

Initializes a new instance of the DiagramLinkCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramLinkCollection Class
MindFusion.Diagramming Namespace

DiagramLinkCollection Methods

DiagramLinkCollection.Clone Method

Creates a shallow copy of this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As DiagramLinkCollection
```

Return Value

An instance of the DiagramLinkCollection class.

See Also

DiagramLinkCollection Members
DiagramLinkCollection Class
MindFusion.Diagramming Namespace

DiagramLinkProperties Class

Stores the values of DiagramLink properties during undo / redo operations.

Syntax

C# Copy Code

```
public class DiagramLinkProperties : DiagramItemProperties
```

Visual Basic Copy Code

```
Public Class DiagramLinkProperties
    Inherits DiagramItemProperties
```

Remarks

If you need to implement undo support for the additional properties defined by a class derived from DiagramLink, do so by deriving from **DiagramLinkProperties** and overriding the SaveProperties, RestoreProperties and CreateProperties methods of **DiagramLink**.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItemProperties
    MindFusion.Diagramming.DiagramLinkProperties

See Also

DiagramLinkProperties Members
MindFusion.Diagramming Namespace

DiagramLinkProperties Members

Stores the values of DiagramLink properties during undo / redo operations.

The following tables list the members exposed by the DiagramLinkProperties type.

Public Constructors

	Name	Description
	DiagramLinkProperties	Initializes a new instance of the **DiagramLinkProperties** class.

Public Fields

	Name	Description
	AllowMoveEnd	For internal use.
	AllowMoveStart	For internal use.
	AutoRoute	For internal use.
	AutoSnapToNode	For internal use.
	BaseShape	For internal use.

	BaseShapeSize	For internal use.
	CascadeOrientation	For internal use.
	CrossingRadius	For internal use.
	DrawCrossings	For internal use.
	Dynamic	For internal use.
	HeadShape	For internal use.
	HeadShapeSize	For internal use.
	IntermediateShape	For internal use.
	IntermediateShapeSize	For internal use.
	ItemClass	For internal use. (Inherited from DiagramItemProperties.)
	labels	For internal use.
	RetainForm	For internal use.
	Selected	For internal use. (Inherited from DiagramItemProperties.)
	Text	For internal use.
	TextPadding	For internal use. (Inherited from DiagramItemProperties.)
	ZIndex	For internal use. (Inherited from DiagramItemProperties.)

Public Methods

	Name	Description
	FreeResources	Frees stored resources which will no longer be used. (Inherited from DiagramItemProperties.)

See Also

DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties Fields

DiagramLinkProperties.AllowMoveEnd Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowMoveEnd
```

Visual Basic Copy Code

```
Public AllowMoveEnd As Boolean
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.AllowMoveStart Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowMoveStart
```

Visual Basic Copy Code

```
Public AllowMoveStart As Boolean
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.AutoRoute Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoRoute
```

Visual Basic Copy Code

```
Public AutoRoute As Boolean
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.AutoSnapToNode Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoSnapToNode
```

Visual Basic Copy Code

```
Public AutoSnapToNode As Boolean
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.BaseShape Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape BaseShape
```

Visual Basic Copy Code

```
Public BaseShape As Shape
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.BaseShapeSize Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double BaseShapeSize
```

Visual Basic Copy Code

```
Public BaseShapeSize As Double
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.CascadeOrientation Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Orientation CascadeOrientation
```

Visual Basic Copy Code

```
Public CascadeOrientation As Orientation
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.CrossingRadius Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double CrossingRadius
```

Visual Basic Copy Code

```
Public CrossingRadius As Double
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.DrawCrossings Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool DrawCrossings
```

Visual Basic Copy Code

```
Public DrawCrossings As Boolean
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.Dynamic Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Dynamic
```

Visual Basic Copy Code

```
Public Dynamic As Boolean
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.HeadShape Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape HeadShape
```

Visual Basic Copy Code

```
Public HeadShape As Shape
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.HeadShapeSize Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double HeadShapeSize
```

Visual Basic Copy Code

```
Public HeadShapeSize As Double
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.IntermediateShape Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape IntermediateShape
```

Visual Basic Copy Code

```
Public IntermediateShape As Shape
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.IntermediateShapeSize Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double IntermediateShapeSize
```

Visual Basic Copy Code

```
Public IntermediateShapeSize As Double
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.labels Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<LinkLabel> labels
```

Visual Basic Copy Code

```
Public labels As List(Of LinkLabel)
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.RetainForm Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RetainForm
```

Visual Basic Copy Code

```
Public RetainForm As Boolean
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties.Text Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Text
```

Visual Basic Copy Code

```
Public Text As String
```

See Also

DiagramLinkProperties Members
DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkProperties Constructor

Initializes a new instance of the DiagramLinkProperties class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkProperties ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramLinkProperties Class
MindFusion.Diagramming Namespace

DiagramLinkState Class

Contains a snapshot of the internal state of a DiagramLink.

Syntax

C# Copy Code

```
public class DiagramLinkState : DiagramItemState
```

Visual Basic Copy Code

```
Public Class DiagramLinkState
    Inherits DiagramItemState
```

Remarks

Instances of this class are used to store the state of links for later undo or redo operations. If you derive from DiagramLink and need to implement undo support for the additional internal state defined by your class, do so by deriving from **DiagramLinkState** and overriding the CreateState, SaveState and RestoreState methods of **DiagramLink**.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItemState
    MindFusion.Diagramming.DiagramLinkState

See Also

DiagramLinkState Members
MindFusion.Diagramming Namespace

DiagramLinkState Members

Contains a snapshot of the internal state of a DiagramLink.

The following tables list the members exposed by the DiagramLinkState type.

Public Constructors

	Name	Description
	DiagramLinkState	Initializes a new instance of the **DiagramLinkState** class.

See Also

DiagramLinkState Class
MindFusion.Diagramming Namespace

DiagramLinkState Constructor

Initializes a new instance of the DiagramLinkState class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkState ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramLinkState Class
MindFusion.Diagramming Namespace

DiagramLinkStyle Class

Represents a reusable set of properties that define the appearance of diagram links.

Syntax

C# Copy Code

```
public class DiagramLinkStyle : DiagramItemStyle
```

Visual Basic Copy Code

```
Public Class DiagramLinkStyle
    Inherits DiagramItemStyle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle
      MindFusion.Diagramming.DiagramLinkStyle

See Also

DiagramLinkStyle Members
MindFusion.Diagramming Namespace

DiagramLinkStyle Members

Represents a reusable set of properties that define the appearance of diagram links.

The following tables list the members exposed by the DiagramLinkStyle type.

Public Constructors

	Name	Description
	DiagramLinkStyle	Initializes a new instance of the **DiagramLinkStyle** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of the item. (Inherited from DiagramItemStyle.)
	FontFamily	Gets or sets the font family of the item text. (Inherited from DiagramItemStyle.)
	FontSize	Gets or sets the font size of the item text. (Inherited from DiagramItemStyle.)
	FontStyle	Gets or sets the font style of the item text. (Inherited from DiagramItemStyle.)
	FontUnit	Gets or sets the font unit of the item text. (Inherited from DiagramItemStyle.)
	HeadStroke	Gets or sets the brush used to paint the arrowhead outlines of this brush.
	HeadStrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline arrowheads.
	HeadStrokeThickness	Gets or sets the outline thickness of the arrowheads of this brush, expressed in the current units of measure.
	ShadowBrush	Gets or sets the brush used to paint the shadow of the item. (Inherited from DiagramItemStyle.)
	Stroke	Gets or sets the brush used to paint the outline of the item. (Inherited from DiagramItemStyle.)
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items. (Inherited from DiagramItemStyle.)
	StrokeThickness	Gets or sets the thickness of the item's outline, expressed in the current units of measure. (Inherited from DiagramItemStyle.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItemStyle.)

Public Methods

	Name	Description
	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
S	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JsonObject.
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)
	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a JsonObject.

	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

DiagramLinkStyle Class
MindFusion.Diagramming Namespace

DiagramLinkStyle Constructor

Initializes a new instance of the DiagramLinkStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkStyle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramLinkStyle Class
MindFusion.Diagramming Namespace

DiagramLinkStyle Methods

DiagramLinkStyle.LoadFromJson Method

Loads the style content from a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
```

```
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the style's data.

context
A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramLinkStyle Members
DiagramLinkStyle Class
MindFusion.Diagramming Namespace

DiagramLinkStyle.SaveToJson Method

Saves the style content into a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject the style's data should be stored into.

context
A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramLinkStyle Members
DiagramLinkStyle Class
MindFusion.Diagramming Namespace

DiagramLinkStyle Properties

DiagramLinkStyle.HeadStroke Property

Gets or sets the brush used to paint the arrowhead outlines of this brush.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush HeadStroke { get; set; }
```

Visual Basic Copy Code

```
Public Property HeadStroke As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

DiagramLinkStyle Members
DiagramLinkStyle Class
MindFusion.Diagramming Namespace

DiagramLinkStyle.HeadStrokeDashStyle Property

Gets or sets the pattern of dashes and gaps used to outline arrowheads.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DashStyle HeadStrokeDashStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property HeadStrokeDashStyle As DashStyle
```

Property Value

A member of the DashStyle enumeration.

See Also

DiagramLinkStyle Members
DiagramLinkStyle Class
MindFusion.Diagramming Namespace

DiagramLinkStyle.HeadStrokeThickness Property

Gets or sets the outline thickness of the arrowheads of this brush, expressed in the current units of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double HeadStrokeThickness { get; set; }
```

Visual Basic Copy Code

```
Public Property HeadStrokeThickness As Double
```

Property Value

A double value.

See Also

DiagramLinkStyle Events

DiagramNode Class

Represents a diagram node.

Syntax

C# Copy Code

```
public abstract class DiagramNode : DiagramItem, IAnimationItem
```

Visual Basic Copy Code

```
Public MustInherit Class DiagramNode
    Inherits DiagramItem
    Implements IAnimationItem
```

Remarks

DiagramNode objects can represent graph vertices, nodes in organizational or flow diagrams, entities in ER diagrams, and so on. **DiagramNode** is an abstract class. You cannot instantiate it directly, but instead use objects from specific classes such as ShapeNode, TableNode, ContainerNode, CompositeNode. You can also create your own node class by deriving from **DiagramNode** or its descendants.

Use the **Bounds** property to get or change the location and size of a node. To move a node without changing its size, call the Move method. To change only the size of a node, invoke the Resize method.

The main textual content of a node is set using its Text property and can be customized by setting Font and TextFormat. Supplementary labels can be created by calling AddLabel.

Constraints upon how users are allowed to modify a node are set through the **Constraints** and **EnabledHandles** properties. The former specifies constraints on move direction and node size. The latter enables or disables specific manipulation handles of a node.

Relations between nodes are indicated by connecting the nodes with DiagramLink instances. Links are considered directed graph edges, so each node has two collections referring to links: **IncomingLinks** and **OutgoingLinks**. **AnchorPattern** defines to which points of a node incoming or outgoing links can be connected. The **AllowIncomingLinks** property lets you specify whether incoming links can be drawn to a node. The **AllowOutgoingLinks** property lets you do that for outgoing links.

A user is allowed to expand or collapse the tree starting from a node if **Expandable** is set to **true**. The **Expanded** property gets or sets whether the subtree starting from a node is expanded or not.

Nodes can be attached to other items, and be used as simple labels, image-holders or other kinds of UI elements. The AttachTo method creates a Group instance if necessary, and calls the group's appropriate AttachTo* method.

Nodes can be rotated, either interactively using the rotation handle above the node, or programmatically using the RotationAngle property.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.CompositeNode
      MindFusion.Diagramming.ContainerNode
      MindFusion.Diagramming.DummyNode
      MindFusion.Diagramming.FreeFormNode
      MindFusion.Diagramming.ShapeNode
      MindFusion.Diagramming.TableNode
      MindFusion.Diagramming.TreeViewNode
      MindFusion.Diagramming.VideoNode
      MindFusion.Diagramming.Mvc.ControlNode
      MindFusion.Diagramming.CaptionedNode
      MindFusion.Diagramming.CompositeNode
      MindFusion.Diagramming.ControlNode
      MindFusion.Diagramming.DummyNode
      MindFusion.Diagramming.FreeFormNode

MindFusion.Diagramming.ShapeNode
MindFusion.Diagramming.VideoNode

See Also

DiagramNode Members
MindFusion.Diagramming Namespace
NodeLabel Class
DiagramLink Class

DiagramNode Members

Represents a diagram node.

The following tables list the members exposed by the DiagramNode type.

Protected Constructors

	Name	Description
	DiagramNode	Overloaded. Initializes a new instance of the **DiagramNode** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node.
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node.
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node.
	Bounds	Gets or sets the rectangle that defines the position of the diagram node.
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Constraints	Allows applying constraints to the location and size of the node.
	Container	Gets the parent ContainerNode containing this node.
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node.
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node.

	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node.
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded.
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like.
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node.
	Labels	Returns the list of supplementary labels displayed by this node.
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached.
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**.
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm.
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node.
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node.
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)

	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override.
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements.
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type.
	AddLabel	Overloaded. Adds a new label to this node.
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node.
	AttachTo	Overloaded. Attaches this node to another diagram item.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node.
	ContainsPoint	Overloaded.
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point.
	Detach	Detaches this node from its master item.
	DetachManipulator	Removes the specified manipulator from this node.
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)

	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface.
	DrawAnchors	Draws the anchor points of this node.
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item.
	DrawInRect	
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node.
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface.
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node.
	Expand	Shows all nodes and links in the subtree starting from this node.
	GetAllIncomingLinks	Overloaded.
	GetAllLinks	Gets all links connected to this node.
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle.
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point.
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline.
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point.
	GetOutlinePoly	DiagramItem.GetOutlinePoly override.
	GetRepaintRect	DiagramItem.GetRepaintRect override.
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point.
	GetRotationAngle	Gets the rotation angle of this node.
	GetToolTip	DiagramItem.GetToolTip override.
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point.
	HitTestManipulators	DiagramItem.HitTestManipulators override.
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes.
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point.

	Move	Moves the node to the specified location.
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node.
	OnDragOver	Invoked when a diagram item is being dragged over this node.
	OnDropOver	Invoked when a diagram item is dropped over this node.
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed.
	ReassignUnanchoredLinks	Reassigns unanchored links
	RemoveLabel	Removes the specified label from this node.
	Resize	Resizes the node.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node.
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node.
	SetPos	Sets the bounding rectangle of this node without updating connected arrows.
	SetRect	This method is used internally by the **Ruler** control to align nodes.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override.
	AllowModify	DiagramItem.AllowModify override.
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled.
	CompleteCreate	DiagramItem.CompleteCreate override.
	CompleteModify	DiagramItem.CompleteModify override.
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed.

	CreateProperties	DiagramItem.CreateProperties override.
	CreateState	DiagramItem.CreateState override.
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history.
	GetAllOutgoingLinks	Overloaded.
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates.
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle.
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy.
	LoadFromJson	Loads the node from a JSON object.
	LoadFromXml	DiagramItem.LoadFromXml override.
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override.
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override.
	OnModify	Invoked when the user moves or resizes this node.
	OnRemove	DiagramItem.OnRemove override.
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively.
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded.
	OnZoomChanged	DiagramItem.OnZoomChanged override.
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override.
	RestoreState	DiagramItem.RestoreState override.
	SaveProperties	DiagramItem.SaveProperties override.

	SaveState	DiagramItem.SaveState override.
	SaveToJson	Saves the node as a JSON object.
	SaveToXml	Saves the item content into an XML element.
	ScaleElements	Scales the node's elements by specified amount.
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary.
	SetExpanded	DiagramItem.SetExpanded override.
	SetIncomingLinks	Sets the incoming links collection of this node.
	SetOutgoingLinks	Sets the outgoing links collection of this node.
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it.
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points.
	StartCreate	DiagramItem.StartCreate override.
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively.
	UpdateCreate	DiagramItem.UpdateCreate override.
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape.
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape.
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item.

See Also

DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode Constructor

Initializes a new instance of the DiagramNode class.

Overload List

Name	Description
DiagramNode ()	Initializes a new instance of the **DiagramNode** class.
DiagramNode (Diagram)	Initializes a new instance of the **DiagramNode** class with default values supplied form the specified Diagram.
DiagramNode (DiagramNode)	Initializes a new instance of the **DiagramNode** class using the specified diagram item as a prototype.

See Also

DiagramNode Constructor ()

Initializes a new instance of the DiagramNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected DiagramNode ()
```

Visual Basic Copy Code

```
Protected Sub New ()
```

Remarks

This constructor sets the DiagramNode properties to their default values.

See Also

DiagramNode Constructor Overload List
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode Constructor (Diagram)

Initializes a new instance of the DiagramNode class with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected DiagramNode (
    Diagram parent
)
```

Visual Basic Copy Code

```
Protected New ( _
    parent As Diagram _
)
```

Parameters

parent
The Diagram from which to obtain default values for the item properties.

Remarks

This constructor initializes the DiagramNode properties with the values specified in the respective default-value properties of the Diagram object. For more information see **Initial Values of Item Properties**.

See Also

DiagramNode Constructor Overload List
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode Constructor (DiagramNode)

Initializes a new instance of the DiagramNode class using the specified diagram item as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected DiagramNode (
    DiagramNode prototype
)
```

Visual Basic Copy Code

```
Protected New ( _
    prototype As DiagramNode _
)
```

Parameters

prototype
The item to use as a prototype. The properties of the new item are initialized from the property values of this item.

Remarks

This constructor copies the DiagramNode property values from the specified prototype node.

See Also

DiagramNode Constructor Overload List
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode Methods

DiagramNode.AcceptLinks Method

Determines whether the node can be connected with links of the specified type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool AcceptLinks (
    bool outgoing
)
```

Visual Basic Copy Code

```
Public Overridable Function AcceptLinks( _
    outgoing As Boolean _
) As Boolean
```

Parameters

outgoing
true to check for outgoing links; otherwise, **false**.

Return Value

true if the node accepts connections of the specified type; otherwise, **false**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.AddLabel Method

Adds a new label to this node.

Overload List

Name	Description
DiagramNode.AddLabel (NodeLabel)	Adds the specified label to this node.
DiagramNode.AddLabel (String)	Adds a new label to this node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.AddLabel Method (NodeLabel)

Adds the specified label to this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void AddLabel (
    NodeLabel label
)
```

Visual Basic Copy Code

```
Public Sub AddLabel( _
    label As NodeLabel _
)
```

Parameters

label

A NodeLabel instance.

Remarks

An item's main text content is specified via its Text property. Call AddLabel if you need to display additional text labels, such as identifiers or hints to the user.

Position of labels can be specified by calling methods such as SetCornerPosition. You can let users move labels interactively by setting the **Behavior** property of **DiagramView** to MoveLabels.

See Also

AddLabel Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
RemoveLabel Method

DiagramNode.AddLabel Method (String)

Adds a new label to this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeLabel AddLabel (
    string text
)
```

Visual Basic Copy Code

```
Public Function AddLabel( _
    text As String _
) As NodeLabel
```

Parameters

text
The label's text.

Return Value

A NodeLabel instance.

Remarks

An item's main text content is specified via its Text property. Call AddLabel if you need to display additional text labels, such as identifiers or hints to the user.

Position of labels can be specified by calling methods such as SetCornerPosition. You can let users move labels interactively by setting the **Behavior** property of **DiagramView** to MoveLabels.

See Also

AddLabel Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
RemoveLabel Method

DiagramNode.AllowCreate Method

DiagramItem.AllowCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override bool AllowCreate (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Function AllowCreate( _
    current As Point, _
    ist As InteractionState _
) As Boolean
```

Parameters

current
The current mouse pointer position.

ist
An InteractionState object containing more information about the current state of the operation.

Return Value

true to allow the creation of this node; otherwise, **false**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.AllowModify Method

DiagramItem.AllowModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override bool AllowModify (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Function AllowModify( _
    current As Point, _
    ist As InteractionState _
) As Boolean
```

Parameters

current
The current mouse pointer position.

ist
An InteractionState object containing more information about the current state of the operation.

Return Value

true to allow the modification of this node; otherwise, **false**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.AttachManipulator Method

Adds the specified manipulator to this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void AttachManipulator (
    Manipulator manipulator
)
```

Visual Basic

```
Public Sub AttachManipulator( _
    manipulator As Manipulator _
)
```

Parameters

manipulator
A **Manipulator** object representing the manipulator to add.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.AttachTo Method

Attaches this node to another diagram item.

Overload List

Name	Description
DiagramNode.AttachTo (DiagramLink, AttachToLink, Int32)	Attaches this node to the specified link using the specified attaching type.
DiagramNode.AttachTo (DiagramNode, AttachToNode)	Attaches this node to the specified node using the specified attaching type.
DiagramNode.AttachTo (DiagramNode, Double, Double)	Attaches this node to the specified main node, keeping original size intact and maintaining original distance to point specified as fraction of main node's size.
DiagramNode.AttachTo (DiagramNode, Double, Double, Double, Double)	Proportionally attaches this node to the specified node.
DiagramNode.AttachTo (DiagramNode, GroupAnchorStyles)	Attaches this node to one or more edges of the specified node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.AttachTo Method (DiagramLink, AttachToLink, Int32)

Attaches this node to the specified link using the specified attaching type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void AttachTo (
    DiagramLink link,
    AttachToLink attType,
    int index
)
```

Visual Basic

```
Public Sub AttachTo( _
    link As DiagramLink, _
    attType As AttachToLink, _
    index As Integer _
)
```

Parameters

link
A DiagramLink to whose control point or segment to attach this node.

attType
A value of the AttachToLink enumeration specifying whether to attach the node to a segment or to a control point of the link.

index
An index of link's control point or segment, depending on the *attType* value.

Remarks

This method attaches the node to a control point or segment of the specified link. If there isn't a Group instance already associated with the master item, one is created automatically by this method. The group is then accessible through the master's SubordinateGroup property, or this node's MasterGroup property.

The node is attached by calling the AttachToLinkPoint or AttachToLinkSegment methods of **Group**, depending on the value of the *attType* argument. The attached node then maintains its original offset from a link's point or segment when link's shape changes.

See Also

AttachTo Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Detach Method

DiagramNode.AttachTo Method (DiagramNode, AttachToNode)

Attaches this node to the specified node using the specified attaching type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void AttachTo (
    DiagramNode node,
    AttachToNode attType
)
```

Visual Basic Copy Code

```
Public Sub AttachTo( _
    node As DiagramNode, _
    attType As AttachToNode _
)
```

Parameters

node
A reference to the node to whose corner to attach this node.

attType
A member of the AttachToNode enumeration specifying to which corner of a node to attach this node.

Remarks

This method attaches the current node to the corner of another node. If there isn't a Group instance already associated with the master item, one is created automatically by this method. The group is then accessible by the master's SubordinateGroup property, or this node's MasterGroup property.

The node is attached by calling the AttachToCorner method of **Group**. The attached node then maintains its original offset from another node's corner point.

See Also

AttachTo Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Detach Method

DiagramNode.AttachTo Method (DiagramNode, Double, Double)

Attaches this node to the specified main node, keeping original size intact and maintaining original distance to point specified as fraction of main node's size.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void AttachTo (
    DiagramNode node,
    float percentX,
    float percentY
)
```

Visual Basic Copy Code

```
Public Sub AttachTo( _
    node As DiagramNode, _
    percentX As Single, _
    percentY As Single _
)
```

Parameters

node
A reference to the node to which to attach this node.

percentX
Percentage of main node's width.

percentY
Percentage of main node's height.

See Also

AttachTo Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.AttachTo Method (DiagramNode, Double, Double, Double, Double)

Proportionally attaches this node to the specified node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void AttachTo (
    DiagramNode node,
    double percentX1,
    double percentY1,
```

```
    double percentX2,
    double percentY2
)
```

Visual Basic Copy Code

```
Public Sub AttachTo( _
    node As DiagramNode, _
    percentX1 As Double, _
    percentY1 As Double, _
    percentX2 As Double, _
    percentY2 As Double _
)
```

Parameters

node
A reference to the node to which to attach this node.

percentX1
Left-hand side coordinate of the node, expressed as percentage of main node's width.

percentY1
Top side coordinate of the node, expressed as percentage of main node's height.

percentX2
Right-hand side coordinate of the node, expressed as percentage of main node's width.

percentY2
Bottom side coordinate of the node, expressed as percentage of main node's height.

Remarks

This method attaches proportionally the current node to another node. If there isn't a Group instance already associated with the master item, one is created automatically by this method. The group is then accessible through the master's SubordinateGroup property, or this node's MasterGroup property.

The node is attached by calling the AttachProportional method of **Group**. The attached node maintains its size and position so that it always covers the specified rectangular area of the master node.

See Also

AttachTo Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Detach Method

DiagramNode.AttachTo Method (DiagramNode, GroupAnchorStyles)

Attaches this node to one or more edges of the specified node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void AttachTo (
    DiagramNode node,
    GroupAnchorStyles anchorStyles
)
```

Visual Basic Copy Code

```
Public Sub AttachTo( _
    node As DiagramNode, _
    anchorStyles As GroupAnchorStyles _
)
```

Parameters

node
A reference to the node to which to attach this node.

anchorStyles
A member of the GroupAnchorStyles enumeration specifying the master node's edges to which this node must be anchored.

Remarks

This method attaches the current node to the corner of another node. If there isn't a Group instance already associated with the master item, one is created automatically by this method. The group is then accessible by the master's SubordinateGroup property, or this node's MasterGroup property.

The node is attached by calling the AttachToEdges method of **Group**, which mimics the anchoring of child controls in a form to the form edges, as implemented in Windows Forms.

See Also

AttachTo Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Detach Method

DiagramNode.CancelModify Method

DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void CancelModify (
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub CancelModify( _
    ist As InteractionState _
)
```

Parameters

ist
An InteractionState object containing more information about the current state of the operation.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.Collapse Method

Hides all nodes and links in the subtree starting from this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Collapse ()
```

Visual Basic

```
Public Sub Collapse ()
```

Remarks

This method is useful mostly when nodes form a tree-like hierarchical structure. Invoking **Collapse** on a node makes all its children in the tree invisible. It depends on the diagram's ExpandOnIncoming property which items connected to the node are considered its children.

Collapse is meant for use with tree structures and it is your responsibility to ensure that your chart is really a tree. A graph is a tree if every node has no more than one parent and there's one root node that can be reached from all nodes going in child-to-parent direction. The tree can be expanded again by calling the Expand method.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Expand Method
Expanded Property

DiagramNode.CompleteCreate Method

DiagramItem.CompleteCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void CompleteCreate (
    Point end
)
```

Visual Basic

```
Protected Overrides Sub CompleteCreate( _
    end As Point _
)
```

Parameters

end
The mouse pointer position.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.CompleteModify Method

DiagramItem.CompleteModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void CompleteModify (
    Point end,
```

```
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub CompleteModify( _
    end As Point, _
    ist As InteractionState _
)
```

Parameters

end

The mouse pointer position.

ist

An InteractionState object containing more information about the current state of the operation.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.ConstraintsInfringed Method

Check whether the KeepInsideParent constraint is being infringed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected bool ConstraintsInfringed ()
```

Visual Basic Copy Code

```
Protected Function ConstraintsInfringed () As Boolean
```

Return Value

true if the current node coordinates violate the KeepInsideParent constraint; otherwise, **false**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.ContainsPoint Method

Overload List

Name	Description
DiagramNode.ContainsPoint (Point)	DiagramItem.ContainsPoint override.
DiagramNode.ContainsPoint (Point, Double)	Tests whether the specified point is contained within the outline of this node, relaxed with the specified threshold.

See Also

DiagramNode.ContainsPoint Method (Point)

DiagramItem.ContainsPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool ContainsPoint (
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function ContainsPoint( _
    point As Point _
) As Boolean
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** value specifying the logical coordinates of a diagram point.

Return Value

true if this node contains the specified point; otherwise, **false**.

See Also

ContainsPoint Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.ContainsPoint Method (Point, Double)

Tests whether the specified point is contained within the outline of this node, relaxed with the specified threshold.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool ContainsPoint (
    Point point,
    double threshold
)
```

Visual Basic Copy Code

```
Public Overrides Function ContainsPoint( _
    point As Point, _
    threshold As Double _
) As Boolean
```

Parameters

point
The point to check.

threshold

Specifies the size of the area around the node outlines where the point is still considered inside the node.

Return Value

true if this node contains the specified point; otherwise, **false**.

See Also

ContainsPoint Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.CreateConnectionPoint Method

Creates a connection point between this node and the specified link at the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual ConnectionPoint CreateConnectionPoint (
    DiagramLink link,
    Point point,
    bool incoming
)
```

Visual Basic Copy Code

```
Public Overridable Function CreateConnectionPoint( _
    link As DiagramLink, _
    point As Point, _
    incoming As Boolean _
) As ConnectionPoint
```

Parameters

link

A DiagramLink that should be connected to this node.

point

The point at which the link should be connected.

incoming

true if *link* is an incoming link, otherwise, **false**.

Return Value

A ConnectionPoint instance representing the connection.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.CreateProperties Method

DiagramItem.CreateProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemProperties CreateProperties ()
```

Visual Basic Copy Code

```
Protected Overrides Function CreateProperties () As DiagramItemProperties
```

Return Value

A DiagramNodeProperties object that will be used to store the properties of this node.

Remarks

This method is invoked by the undo/redo subsystem when creating a ChangeItemCmd instance.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
SaveProperties Method
RestoreProperties Method

DiagramNode.CreateState Method

DiagramItem.CreateState override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemState CreateState ()
```

Visual Basic Copy Code

```
Protected Overrides Function CreateState () As DiagramItemState
```

Return Value

A DiagramNodeState instance.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.Detach Method

Detaches this node from its master item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Detach ()
```

Visual Basic Copy Code

```
Public Sub Detach ()
```

Remarks

The node should be previously attached using AttachTo.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.DetachManipulator Method

Removes the specified manipulator from this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void DetachManipulator (
    Manipulator manipulator
)
```

Visual Basic Copy Code

```
Public Sub DetachManipulator( _
    manipulator As Manipulator _
)
```

Parameters

manipulator
A **Manipulator** object representing the manipulator to remove.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.Draw Method

DiagramItem.Draw override. Draws the item on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Draw (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub Draw( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
The IGraphics surface to draw the node on.

options
Provides options for the drawing operation.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.DrawAnchors Method

Draws the anchor points of this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void DrawAnchors (
    IGraphics graphics
)
```

Visual Basic Copy Code

```
Public Overridable Sub DrawAnchors( _
    graphics As IGraphics _
)
```

Parameters

graphics
The IGraphics surface to draw the anchor points on.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.DrawHandles Method

DiagramItem.DrawHandles override. Renders the adjustment handles of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawHandles (
    IGraphics graphics,
    HandlesVisualStyle style,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawHandles( _
    graphics As IGraphics, _
    style As HandlesVisualStyle, _
    options As RenderOptions _
)
```

Parameters

graphics
The IGraphics surface on which the handles should be drawn.

style
A HandlesVisualStyle instance containing brush and pen objects that should be used when drawing handles.

options
Provides options for the drawing operation.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
HitTestHandle Method
HandlesStyle Property

DiagramNode.DrawInRect Method

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawInRect (
    IGraphics graphics,
    RenderOptions options,
    Rect target
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawInRect( _
    graphics As IGraphics, _
    options As RenderOptions, _
    target As Rect _
)
```

Parameters

graphics

options

target

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.DrawLocal Method

Override in a derived class to provide implementation specific drawing for the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void DrawLocal (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overridable Sub DrawLocal( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
The IGraphics surface to draw the node on.

options
Provides options for the drawing operation.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.DrawShadow Method

DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawShadow (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawShadow( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
An IGraphics object where the shadow should be rendered.

options
Provides options for the drawing operation.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.DrawShadowLocal Method

Override in a derived class to provide implementation specific drawing for the shadow of the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void DrawShadowLocal (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overridable Sub DrawShadowLocal( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
An IGraphics object where the shadow should be rendered.

options
Provides options for the drawing operation.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.Expand Method

Shows all nodes and links in the subtree starting from this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Expand ()
```

Visual Basic Copy Code

```
Public Sub Expand ()
```

Remarks

This method is useful mostly when nodes form a tree-like hierarchical structure. Use **Expand** to show a subtree of nodes that was previously collapsed, either interactively or by calling Collapse. If the RecursiveExpand property is enabled, all tree levels are expanded recursively.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Collapse Method
Expanded Property

DiagramNode.FreeResources Method

Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void FreeResources ()
```

Visual Basic Copy Code

```
Protected Overrides Sub FreeResources ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetAllIncomingLinks Method

Overload List

Name	Description
DiagramNode.GetAllIncomingLinks ()	Gets a collection of all links entering this node.
DiagramNode.GetAllIncomingLinks (DiagramLinkCollection)	Adds all links going in this node to the specified collection.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetAllIncomingLinks Method ()

Gets a collection of all links entering this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection GetAllIncomingLinks ()
```

Visual Basic Copy Code

```
Public Function GetAllIncomingLinks () As DiagramLinkCollection
```

Return Value

See Also

GetAllIncomingLinks Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetAllIncomingLinks Method (DiagramLinkCollection)

Adds all links going in this node to the specified collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void GetAllIncomingLinks (
    DiagramLinkCollection links
)
```

Visual Basic Copy Code

```
Protected Overridable Sub GetAllIncomingLinks( _
    links As DiagramLinkCollection _
)
```

Parameters

links

See Also

GetAllIncomingLinks Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetAllLinks Method

Gets all links connected to this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection GetAllLinks ()
```

Visual Basic Copy Code

```
Public Function GetAllLinks () As DiagramLinkCollection
```

Return Value

A DiagramLinkCollection containing the links connected to this node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetAllOutgoingLinks Method

Overload List

Name	Description
DiagramNode.GetAllOutgoingLinks ()	Gets a collection of all links going out of this node.
DiagramNode.GetAllOutgoingLinks (DiagramLinkCollection)	Adds all links going out of this node to the specified collection.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetAllOutgoingLinks Method ()

Gets a collection of all links going out of this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection GetAllOutgoingLinks ()
```

Visual Basic Copy Code

```
Public Function GetAllOutgoingLinks () As DiagramLinkCollection
```

Return Value

See Also

GetAllOutgoingLinks Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetAllOutgoingLinks Method (DiagramLinkCollection)

Adds all links going out of this node to the specified collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void GetAllOutgoingLinks (
    DiagramLinkCollection links
)
```

Visual Basic Copy Code

```
Protected Overridable Sub GetAllOutgoingLinks( _
    links As DiagramLinkCollection _
```

```
)
```

Parameters

links

See Also

GetAllOutgoingLinks Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetCenter Method

Gets the center of this node's bounding rectangle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point GetCenter ()
```

Visual Basic Copy Code

```
Public Function GetCenter () As Point
```

Return Value

A **Microsoft.Maui.Graphics.Point** representing the center point of this node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetClosestAnchorPoint Method

Returns the index of the anchor point that lies at the shortest distance from the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int GetClosestAnchorPoint (
    Point point,
    bool incoming
)
```

Visual Basic Copy Code

```
Public Function GetClosestAnchorPoint( _
    point As Point, _
    incoming As Boolean _
) As Integer
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying a location in the diagram.

incoming
true to look for anchor points that allow incoming links; false to look for ones that allow outgoing links.

Return Value

The index of the found point within the AnchorPattern definition.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetIntersection Method

Calculates the intersection point between the segment defined by the specified point pair and the node's outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Point GetIntersection (
    Point segmentStart,
    Point segmentEnd
)
```

Visual Basic Copy Code

```
Public Overridable Function GetIntersection( _
    segmentStart As Point, _
    segmentEnd As Point _
) As Point
```

Parameters

segmentStart
The first point of the segment.

segmentEnd
The second point of the segment.

Return Value

A **Microsoft.Maui.Graphics.Point** representing the intersection point.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetLocalBounds Method

Gets this node's bounding rectangle in local coordinates.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected Rect GetLocalBounds ()
```

Visual Basic Copy Code

```
Protected Function GetLocalBounds () As Rect
```

Return Value

A **Microsoft.Maui.Graphics.Rect** representing the bounding rectangle of this node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetNearestBorderPoint Method

Returns a point that lies on the outline of this node, at the shortest distance from the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual Point GetNearestBorderPoint (
    Point point
)
```

Visual Basic Copy Code

```
Public Overridable Function GetNearestBorderPoint( _
    point As Point _
) As Point
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying a point in the diagram.

Return Value

A **Point** specifying the point from the node's outline closest to the point specified as argument.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetOutlinePoly Method

DiagramItem.GetOutlinePoly override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override PointCollection GetOutlinePoly (
    bool rendering
)
```

Visual Basic Copy Code

```
Public Overrides Function GetOutlinePoly( _
    rendering As Boolean _
) As PointCollection
```

Parameters

rendering

Return Value

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetRepaintRect Method

DiagramItem.GetRepaintRect override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Rect GetRepaintRect (
    bool includeConnected
)
```

Visual Basic Copy Code

```
Public Overrides Function GetRepaintRect( _
    includeConnected As Boolean _
) As Rect
```

Parameters

includeConnected
true if related items should be considered too; otherwise, **false**.

Return Value

A rectangular region that should be repainted when this node is modified.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetRgnInRect Method

Creates and returns a **Region** object for this node as though it is contained in the specified rectangle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual IGraphicsPath GetRgnInRect (
    Rect rc,
    bool inLocal
)
```

Visual Basic Copy Code

```
Protected Overridable Function GetRgnInRect( _
    rc As Rect, _
    inLocal As Boolean _
) As IGraphicsPath
```

Parameters

rc

inLocal

Return Value

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetRotatedBounds Method

Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point.

Overload List

Name	Description
DiagramNode.GetRotatedBounds ()	DiagramItem.GetRotatedBounds override.
DiagramNode.GetRotatedBounds (Rect, Point, Double)	Gets the bounding rectangle of this node, when it is rotated at the specified angle around the specified pivot.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetRotatedBounds Method ()

DiagramItem.GetRotatedBounds override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Rect GetRotatedBounds ()
```

Visual Basic Copy Code

```
Public Overrides Function GetRotatedBounds () As Rect
```

Return Value

A rectangle that bounds the rotated shape of the item.

See Also

GetRotatedBounds Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetRotatedBounds Method (Rect, Point, Double)

Gets the bounding rectangle of this node, when it is rotated at the specified angle around the specified pivot.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected static Rect GetRotatedBounds (
    Rect rect,
    Point center,
    double angle
)
```

Visual Basic Copy Code

```
Protected Shared Function GetRotatedBounds( _
    rect As Rect, _
    center As Point, _
    angle As Double _
) As Rect
```

Parameters

rect
The rectangle that will be rotated.

center
The pivot point.

angle
The rotation angle.

Return Value

A **Microsoft.Maui.Graphics.Rect** representing the new bounding rectangle.

See Also

GetRotatedBounds Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetRotationAngle Method

Gets the rotation angle of this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual double GetRotationAngle ()
```

Visual Basic Copy Code

```
Public Overridable Function GetRotationAngle () As Double
```

Return Value

A **double** value specifying the rotation angle.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GetToolTip Method

DiagramItem.GetToolTip override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override string GetToolTip (
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function GetToolTip( _
    point As Point _
) As String
```

Parameters

point
The mouse pointer position.

Return Value

A string specifying the tooltip text.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.GroupContains Method

Determines whether this node is a predecessor of the specified node in the group hierarchy.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected bool GroupContains (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Protected Function GroupContains( _
    node As DiagramNode _
) As Boolean
```

Parameters

node
The node to check.

Return Value

true if the specified node is subordinate to this one; otherwise, **false**.

See Also

DiagramNode.HitTestHandle Method

DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override AdjustmentHandle HitTestHandle (
    Point point,
    double distance
)
```

Visual Basic Copy Code

```
Public Overrides Function HitTestHandle( _
    point As Point, _
    distance As Double _
) As AdjustmentHandle
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying where to look for a selection handle.

distance
Maximum distance from a handle at which it is still considered as hit.

Return Value

Returns an AdjustmentHandle instance representing the adjustment handle found at the specified location, or **null** if there is none.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
DrawHandles Method
HandlesStyle Property

DiagramNode.HitTestManipulators Method

DiagramItem.HitTestManipulators override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Manipulator HitTestManipulators (
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function HitTestManipulators( _
    point As Point _
) As Manipulator
```

Parameters

point

Return Value

An instance of **Manipulator** -derived class, or **null** if there isn't one found at specified location.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.Intersects Method

Determines whether the node intersects another one, taking into account the outlines of node shapes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Intersects (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Function Intersects( _
    node As DiagramNode _
) As Boolean
```

Parameters

node
A reference to the node to test for intersections with this one.

Return Value

true if the current node intersects with the specified node; otherwise, **false**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Bounds Property

DiagramNode.LabelFromPoint Method

Returns the label whose layout rectangle contains the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override ItemLabel LabelFromPoint (
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function LabelFromPoint( _
    point As Point _
```

```
) As ItemLabel
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** instance.

Return Value

The NodeLabel that contains the specified point.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.LoadFromJson Method

Loads the node from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the node's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

Remarks

This method is invoked when an item is being deserialized from a diagram file saved in JSON format.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.LoadFromXml Method

DiagramItem.LoadFromXml override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** containing the item's data.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.Move Method

Moves the node to the specified location.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Move (
    double x,
    double y
)
```

Visual Basic Copy Code

```
Public Sub Move( _
    x As Double, _
    y As Double _
)
```

Parameters

x

A **double** value specifying the new X-coordinate of node's location.

y

A **double** value specifying the new Y-coordinate of node's location.

Remarks

This method moves a node to the specified position, without changing the node's size. Moving a node also updates the control points coordinates of all links connected to the node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Bounds Property
Resize Method

DiagramNode.OnAdd Method

DiagramItem.OnAdd override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnAdd ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnAdd ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.OnDragOut Method

Invoked when a diagram item is being dragged out of this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void OnDragOut (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Overridable Sub OnDragOut( _
    item As DiagramItem _
)
```

Parameters

item
The DiagramItem that is dragged out of the node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.OnDragOver Method

Invoked when a diagram item is being dragged over this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool OnDragOver (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Overridable Function OnDragOver( _
    item As DiagramItem _
) As Boolean
```

Parameters

item

The DiagramItem that is dragged over the node.

Return Value

false if the diagram should notify nodes below this one; otherwise, **true**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.OnDropOver Method

Invoked when a diagram item is dropped over this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual bool OnDropOver (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Overridable Function OnDropOver( _
    item As DiagramItem _
) As Boolean
```

Parameters

item

The DiagramItem that is dropped over the node.

Return Value

false if the diagram should notify nodes below this one; otherwise, **true**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.OnMeasureUnitChanged Method

DiagramItem.OnMeasureUnitChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void OnMeasureUnitChanged ()
```

Visual Basic

```
Protected Overrides Sub OnMeasureUnitChanged ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.OnModify Method

Invoked when the user moves or resizes this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected virtual void OnModify (
    Point mousePosition,
    AdjustmentHandles adjustmentHandle
)
```

Visual Basic

```
Protected Overridable Sub OnModify( _
    mousePosition As Point, _
    adjustmentHandle As AdjustmentHandles _
)
```

Parameters

mousePosition
Specifies the mouse position.

adjustmentHandle
Specifies the adjustment handle used to modify the node.

Remarks

Implement this method in a custom node class if you need to detect modifications done by the user without handling events of the Diagram class.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.OnRemove Method

DiagramItem.OnRemove override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void OnRemove ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnRemove ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.OnRotationAngleChanged Method

Invoked when the rotation angle of this node has been changed, either programmatically or interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnRotationAngleChanged ()
```

Visual Basic Copy Code

```
Protected Overridable Sub OnRotationAngleChanged ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.OnUpdateBounds Method

Overload List

Name	Description
DiagramNode.OnUpdateBounds ()	Invoked when the bounding rectangle of the node has changed.
DiagramNode.OnUpdateBounds (Boolean)	Invoked when the bounding rectangle of the node has changed.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.OnUpdateBounds Method ()

Invoked when the bounding rectangle of the node has changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnUpdateBounds ()
```

Visual Basic Copy Code

```
Protected Overridable Sub OnUpdateBounds ()
```

See Also

OnUpdateBounds Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.OnUpdateBounds Method (Boolean)

Invoked when the bounding rectangle of the node has changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnUpdateBounds (
    bool sizeChanged
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnUpdateBounds( _
    sizeChanged As Boolean _
)
```

Parameters

sizeChanged
true if the size has changed; **false** if only the position has changed.

See Also

OnUpdateBounds Method Overload List
DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.OnZoomChanged Method

DiagramItem.OnZoomChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnZoomChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnZoomChanged ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.ReassignAnchorPoints Method

Reassigns links to anchor points so that they are evenly distributed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ReassignAnchorPoints ()
```

Visual Basic Copy Code

```
Public Sub ReassignAnchorPoints ()
```

Remarks

If there are more anchor points defined for this node than links connected to it, this method ensures that each point has at most one link attached to it. **ReassignAnchorPoints** also tries to assign links to points that are close to their opposite end points.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
AnchorPattern Property

DiagramNode.ReassignUnanchoredLinks Method

Reassigns unanchored links

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ReassignUnanchoredLinks ()
```

Visual Basic Copy Code

```
Public Sub ReassignUnanchoredLinks ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.RemoveLabel Method

Removes the specified label from this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RemoveLabel (
    NodeLabel label
)
```

Visual Basic Copy Code

```
Public Sub RemoveLabel( _
    label As NodeLabel _
)
```

Parameters

label

A NodeLabel instance.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
AddLabel Method

DiagramNode.Resize Method

Resizes the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Resize (
    double width,
    double height
)
```

Visual Basic Copy Code

```
Public Sub Resize( _
    width As Double, _
    height As Double _
)
```

Parameters

width

A **double** value specifying new width of the node.

height

A **double** value specifying new height of the node.

Remarks

This method resizes a node, without changing the node's top-left corner location. Resizing a node also updates the control points coordinates of all links connected to the node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Bounds Property
Move Method

DiagramNode.RestoreProperties Method

DiagramItem.RestoreProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties -derived object containing a snapshot of the item's properties.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.RestoreState Method

DiagramItem.RestoreState override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreState (
    DiagramItemState state
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreState( _
    state As DiagramItemState _
)
```

Parameters

state
A DiagramItemState -derived object.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SaveProperties Method

DiagramItem.SaveProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props

A DiagramItemProperties -derived object where the item's properties should be saved.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SaveState Method

DiagramItem.SaveState override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveState (
    DiagramItemState state
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveState( _
    state As DiagramItemState _
)
```

Parameters

state

A DiagramItemState-derived object.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SaveToJson Method

Saves the node as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject instance to contain the serialized data.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

Remarks

This method is invoked when an item is being serialized into a JSON-format diagram file.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SaveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** the item's data should be stored into.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

DiagramNode.ScaleElements Method

Scales the node's elements by specified amount.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void ScaleElements (
    double scale
)
```

Visual Basic Copy Code

```
Protected Overridable Sub ScaleElements( _
    scale As Double _
)
```

Parameters

scale
The value by which to multiply node elements' sizes.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SetBounds Method

Sets the location and size of this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetBounds (
    Rect rect,
    bool updateGroups,
    bool updateLinks
)
```

Visual Basic Copy Code

```
Public Sub SetBounds( _
    rect As Rect, _
    updateGroups As Boolean, _
    updateLinks As Boolean _
)
```

Parameters

rect
A **Microsoft.Maui.Graphics.Rect** specifying the node position.

updateGroups
Specifies whether the position of child nodes attached to this one should be updated too.

updateLinks
Specifies whether the position of links connected to this node should be updated too.

Remarks

This method lets you set the position of a node without affecting items related to it, such as its incoming and outgoing links, or nodes set as its children using AttachTo. Assigning to the **Bounds** property or calling the Move and Resize methods always updates the related items.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Bounds Property

DiagramNode.SetExpandable Method

Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SetExpandable (
    bool exp
)
```

Visual Basic Copy Code

```
Protected Sub SetExpandable( _
    exp As Boolean _
)
```

Parameters

exp
true if a collapse/expand button should be added to this node; otherwise, **false**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SetExpanded Method

DiagramItem.SetExpanded override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SetExpanded (
    bool e
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SetExpanded( _
    e As Boolean _
```

```
)
```

Parameters

e

Specifies the new state of the node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SetExpandedFlag Method

Sets the 'expanded' flag, repainting the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetExpandedFlag (
    bool newState
)
```

Visual Basic Copy Code

```
Public Sub SetExpandedFlag( _
    newState As Boolean _
)
```

Parameters

newState

Specifies the new state of the node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SetIncomingLinks Method

Sets the incoming links collection of this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SetIncomingLinks (
    DiagramLinkCollection links
)
```

Visual Basic Copy Code

```
Protected Sub SetIncomingLinks( _
    links As DiagramLinkCollection _
)
```

Parameters

links
A DiagramLinkCollection containing the node's incoming links.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SetOutgoingLinks Method

Sets the outgoing links collection of this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SetOutgoingLinks (
    DiagramLinkCollection links
)
```

Visual Basic Copy Code

```
Protected Sub SetOutgoingLinks( _
    links As DiagramLinkCollection _
)
```

Parameters

links
A DiagramLinkCollection containing the node's outgoing links.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SetPos Method

Sets the bounding rectangle of this node without updating connected arrows.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void SetPos (
    Nullable<Single> x,
    Nullable<Single> y,
    Nullable<Single> w,
    Nullable<Single> h
)
```

Visual Basic Copy Code

```
Public Overridable Sub SetPos( _
    x As Nullable(Of Single), _
    y As Nullable(Of Single), _
    w As Nullable(Of Single), _
    h As Nullable(Of Single) _
)
```

Parameters

x

y

w

h

Remarks

For internal use.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SetRect Method

This method is used internally by the **Ruler** control to align nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetRect (
    Rect rect,
    bool raiseModfEvent
)
```

Visual Basic Copy Code

```
Public Sub SetRect( _
    rect As Rect, _
    raiseModfEvent As Boolean _
)
```

Parameters

rect
The new location of the node.

raiseModfEvent
A flag specifying whether to raise the NodeModified event.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.SetReflexive Method

Invoked to let a node set the shape of a reflexive link connected to it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SetReflexive (
    DiagramLink link
```

```
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SetReflexive( _
    link As DiagramLink _
)
```

Parameters

link
Reference to the link.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.ShouldRenderAnchors Method

Determines whether the node should render its anchor points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected bool ShouldRenderAnchors (
    RenderOptions options
)
```

Visual Basic Copy Code

```
Protected Function ShouldRenderAnchors( _
    options As RenderOptions _
) As Boolean
```

Parameters

options
A RenderOptions object providing more information about the current drawing context.

Return Value

true if anchor points should be rendered; otherwise, **false**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.StartCreate Method

DiagramItem.StartCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void StartCreate (
    Point org
)
```

Visual Basic Copy Code

```
Protected Overrides Sub StartCreate( _
    org As Point _
)
```

Parameters

org
Position of the mouse pointer.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.StartModify Method

DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void StartModify (
    Point org,
    AdjustmentHandle handle,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub StartModify( _
    org As Point, _
    handle As AdjustmentHandle, _
    ist As InteractionState _
)
```

Parameters

org
Position of the mouse pointer.

handle
The adjustment handle used to modify this node.

ist
An InteractionState object containing more information about the current state of the operation.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.UpdateAnimation Method

Invoked by running animations to update an item according to current animation progress.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void UpdateAnimation (
    Animation animation,
    double animationDelta
)
```

Visual Basic Copy Code

```
Public Overridable Sub UpdateAnimation( _
    animation As Animation, _
    animationDelta As Double _
)
```

Parameters

animation
The Animation object used to animate this node.

animationDelta
A value representing the current animation progress

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.UpdateCreate Method

DiagramItem.UpdateCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void UpdateCreate (
    Point current
)
```

Visual Basic Copy Code

```
Protected Overrides Sub UpdateCreate( _
    current As Point _
)
```

Parameters

current
Position of the mouse pointer.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.UpdateLinkIntersections Method

Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void UpdateLinkIntersections ()
```

Visual Basic Copy Code

```
Protected Sub UpdateLinkIntersections ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.UpdateLinkPositions Method

Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void UpdateLinkPositions ()
```

Visual Basic Copy Code

```
Protected Sub UpdateLinkPositions ()
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.UpdateModify Method

DiagramItem.UpdateModify override. Invoked during interactive modification of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void UpdateModify (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub UpdateModify( _
    current As Point, _
    ist As InteractionState _
)
```

Parameters

current

Position of the mouse pointer.

ist

An InteractionState object containing more information about the current state of the operation.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode Properties

DiagramNode.AllowIncomingLinks Property

Gets or sets a value indicating whether users are allowed to draw incoming links to this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowIncomingLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowIncomingLinks As Boolean
```

Property Value

true to let users draw incoming links to this node; otherwise, **false**. The default is **true**.

Remarks

Links to a node can be created programmatically regardless of the value of this property.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
IncomingLinks Property
AllowOutgoingLinks Property

DiagramNode.AllowOutgoingLinks Property

Gets or sets a value indicating whether users are allowed to draw outgoing links from this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowOutgoingLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowOutgoingLinks As Boolean
```

Property Value

true to let users draw outgoing links from this node; otherwise, **false**. The default is **true**.

Remarks

Links from a node can be created programmatically regardless of the value of this property.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
OutgoingLinks Property
AllowIncomingLinks Property

DiagramNode.AnchorPattern Property

Gets or sets the anchor points to which links are attached when connected to the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern AnchorPattern { get; set; }
```

Visual Basic Copy Code

```
Public Property AnchorPattern As AnchorPattern
```

Property Value

An instance of the AnchorPattern class.

Remarks

This property specifies a set of points to which links can be attached. Each anchor point can accept either incoming links, outgoing links or both. If a node does not have any anchor points defined, links can be connected to any part of the node.

A ShapeNode whose **AnchorPattern** is not explicitly set uses the AnchorPattern from the Shape definition.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Diagram.SnapToAnchor Property
ReassignAnchorPoints Method

DiagramNode.Bounds Property

Gets or sets the rectangle that defines the position of the diagram node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect Bounds { get; set; }
```

Visual Basic Copy Code

```
Public Property Bounds As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** instance.

Remarks

Use this property to set or get the size and/or location of a node. Moving a node also updates the control points coordinates of all links connected to the node. If there are subordinate items attached to it, they are moved too by the translation vector defined by the **Bounds** change. To change the bounding rectangle of a node without affecting related items, call the SetBounds method.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Move Method
Resize Method
Intersects Method

DiagramNode.Constraints Property

Allows applying constraints to the location and size of the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeConstraints Constraints { get; set; }
```

Visual Basic Copy Code

```
Public Property Constraints As NodeConstraints
```

Property Value

An instance of the NodeConstraints class.

Remarks

These constraints are considered when a user modifies the node interactively. They are ignored if node placement is changed programmatically by using methods and properties.

Example

This sample code disables moving or resizing a node vertically:

C# Copy Code

```
DiagramNode node = diagram.ActiveItem as DiagramNode;

// Allow moving the node only horizontally
node.Constraints.MoveDirection = DirectionConstraint.Horizontal;

// Disable handles that can be used to resize the node vertically
node.EnabledHandles = AdjustmentHandles.Move |
    AdjustmentHandles.ResizeMiddleLeft | AdjustmentHandles.ResizeMiddleRight;
```

Visual Basic Copy Code

```
Dim node As DiagramNode = diagram.ActiveItem

' Allow moving the node only horizontally
node.Constraints.MoveDirection = DirectionConstraint.Horizontal

' Disable handles that can be used to resize the node vertically
node.EnabledHandles = AdjustmentHandles.Move + _
    AdjustmentHandles.ResizeMiddleLeft + AdjustmentHandles.ResizeMiddleRight
```

See Also

DiagramNode.Container Property

Gets the parent ContainerNode containing this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNode Container { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Container As ContainerNode
```

Property Value

The parent ContainerNode of this node.

Remarks

Call the Add method of ContainerNode to add a child node to the container. Call Remove to remove a child node.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
MasterItem Property
ContainerNode.Children Property

DiagramNode.Effects Property

Gets a collection with all effects applied to this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IList<NodeEffect> Effects { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Effects As IList(Of NodeEffect)
```

Property Value

A list with all node effects.

Remarks

The local node effects are applied after the diagram-wide node effects defined in NodeEffects. However, if the first effect in this list is **null** (**Nothing** in Visual Basic), the diagram effects are ignored.

See Also

DiagramNode Members
DiagramNode Class

DiagramNode.EnabledHandles Property

Gets or sets the kinds of modifications that end-users are permitted to perform on the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AdjustmentHandles EnabledHandles { get; set; }
```

Visual Basic Copy Code

```
Public Property EnabledHandles As AdjustmentHandles
```

Property Value

A combination of the AdjustmentHandles enumeration members. The default is **AdjustmentHandles.All**.

Remarks

Disabling some selection handles allows restricting the types of modifications that users can perform on a node. Each manipulation handle can be enabled or disabled by setting or clearing the respective flag in the combination.

For example, to make a node resizable only horizontally, set this property to a combination of the Move, **ResizeMiddleLeft** and **ResizeMiddleRight** flags. That will disable all manipulation handles, except the left and right ones, which allow horizontal resizing, and the central one allowing movement of the node.

Example

The following code disables the rotation handle of a shape node:

C# Copy Code

```
node.EnabledHandles = AdjustmentHandles.All & ~ AdjustmentHandles.Rotate;
```

Visual Basic Copy Code

```
node.EnabledHandles = AdjustmentHandles.All And Not AdjustmentHandles.Rotate
```

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Constraints Property
DiagramView.ModificationStart Property
Diagram.DisabledHandlesColor Property
Diagram.ShowDisabledHandles Property

DiagramNode.Expandable Property

Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Expandable { get; set; }
```

Visual Basic

```
Public Property Expandable As Boolean
```

Property Value

true if users are allowed to expand / collapse the tree starting from this node; otherwise, **false**. The default is **false**.

Remarks

This property is useful mostly when nodes form a tree-like hierarchical structure. If it is enabled, a small [±] button is drawn beside the node. Clicking that button either shows or hides all children of this node in the tree. It depends on the diagram's ExpandOnIncoming property which items connected to a node are considered its children. The tree expansion continues recursively down the child nodes if the RecursiveExpand property is enabled.

A tree can be collapsed or expanded programmatically by invoking the Collapse or Expand methods on the tree root node.

The appearance of the collapse / expand icon can be customized by setting **Renderers.ExpandIconRenderer**.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Expanded Property
Diagram.NodesExpandable Property
HtmlBuilder.ExpandBtnHyperLink Property

DiagramNode.Expanded Property

Gets or sets a value indicating whether the subtree starting from this node is expanded.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public virtual bool Expanded { get; set; }
```

Visual Basic

```
Public Overridable Property Expanded As Boolean
```

Property Value

true if the tree starting from this node is expanded; otherwise, **false**.

Remarks

This property is useful mostly when nodes form a tree-like hierarchical structure. Set it to **false** to make the subtree starting from a node collapse, with all its child nodes and links becoming invisible. Set it to **true** to show the child nodes and links again.

A user is allowed to expand or collapse the tree that starts from a node interactively if the **Expandable** property of that node is enabled.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Expandable Property
Expand Method

DiagramNode.HandlesStyle Property

Gets or sets a value indicating how the node adjustment handles behave and what do they look like.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HandlesStyle HandlesStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property HandlesStyle As HandlesStyle
```

Property Value

A member of the HandlesStyle enumeration.

Remarks

Specifies how node selection status is indicated visually. In addition, this property defines how manipulation of the node can be done, that is, how to move it, resize it or link it to other nodes. You can assign any of the HandlesStyle values to this property.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
ShapeHandlesStyle Property
DrawHandles Method
HitTestHandle Method

DiagramNode.HasItems Property

Determines whether the node supports connections to its constituent elements.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual bool HasItems { get; }
```

Visual Basic Copy Code

```
Protected Overridable ReadOnly Property HasItems As Boolean
```

Property Value

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.IncomingLinks Property

Gets a collection of links that point to the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection IncomingLinks { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property IncomingLinks As DiagramLinkCollection
```

Property Value

A reference to a DiagramLinkCollection object.

Remarks

Use this collection to get access to every link pointing to the node, as well as get the number of such links. If the node is a table, links that connect to it as a whole entity are included in the collection, and links that connect to particular row of the table are included in IncomingLinks.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
AllowIncomingLinks Property
OutgoingLinks Property
TableNode.ConnectionStyle Property
Query Method

DiagramNode.Labels Property

Returns the list of supplementary labels displayed by this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<NodeLabel> Labels { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Labels As List(Of NodeLabel)
```

Property Value

A list of NodeLabel objects.

Remarks

Call the AddLabel and RemoveLabel methods to add or remove labels.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Text Property

DiagramNode.MasterGroup Property

Gets a reference to the group to which this node is attached.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Group MasterGroup { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MasterGroup As Group
```

Property Value

A reference to a Group object or **null** (**Nothing** in Visual Basic) if the node is not attached to a group.

Remarks

Diagram items can be grouped hierarchically in such a manner, that if an item is moved all of its subordinated items in the hierarchy move too. The immediate subordinates of an item are contained and represented by a Group instance. This property provides access to the group that contains the item as a subordinate. If the item is not a subordinate of another, the property returns **null** (**Nothing** in Visual Basic).

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
SubordinateGroup Property
Factory.CreateGroup Method
Query Method

DiagramNode.MasterItem Property

Returns the node's Container or main item of MasterGroup.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem MasterItem { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MasterItem As DiagramItem
```

Property Value

A DiagramItem reference indicating master item in containment or group hierarchy.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
Group Class
ContainerNode Class

DiagramNode.Obstacle Property

Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Obstacle { get; set; }
```

Visual Basic Copy Code

```
Public Property Obstacle As Boolean
```

Property Value

true makes routed links look for a path going around this node; **false** lets links cross the node. The default is **true**.

Remarks

It is useful to set this property to **false** for nodes used as decorative containers or labels.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
DiagramLink.AutoRoute Property
Diagram.RouteAllLinks Method

DiagramNode.OutgoingLinks Property

Gets a collection of links that start from this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection OutgoingLinks { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property OutgoingLinks As DiagramLinkCollection
```

Property Value

A reference to a DiagramLinkCollection object.

Remarks

Use this collection to get access to every link starting from the node, as well as get the number of such links. If the node is a table, links that connect to it as a whole entity are included in the collection, and links that connect to particular row of the table are included in OutgoingLinks.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace
AllowOutgoingLinks Property
IncomingLinks Property
TableNode.ConnectionStyle Property
Query Method

DiagramNode.RotationAngle Property

Gets or sets the rotation angle of this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double RotationAngle { get; set; }
```

Visual Basic

```
Public Property RotationAngle As Double
```

Property Value

A **double** value specifying the rotation angle in degrees.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNode.TopLevel Property

DiagramItem.TopLevel override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override bool TopLevel { get; }
```

Visual Basic

```
Public Overrides ReadOnly Property TopLevel As Boolean
```

Property Value

true if this is a top-level item in the diagram; **false** otherwise.

See Also

DiagramNode Members
DiagramNode Class
MindFusion.Diagramming Namespace

DiagramNodeCollection Class

Represents a strongly typed collection of diagram nodes.

Syntax

C#

```
public class DiagramNodeCollection : ItemCollectionBase<DiagramNode>
```

Visual Basic

```
Public Class DiagramNodeCollection
    Inherits ItemCollectionBase(Of DiagramNode)
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.CollectionBase(DiagramNode)
    MindFusion.Diagramming.ItemCollectionBase(DiagramNode)
      MindFusion.Diagramming.DiagramNodeCollection

See Also

DiagramNodeCollection Members
MindFusion.Diagramming Namespace

DiagramNodeCollection Members

Represents a strongly typed collection of diagram nodes.

The following tables list the members exposed by the DiagramNodeCollection type.

Public Constructors

	Name	Description
	DiagramNodeCollection	Initializes a new instance of the **DiagramNodeCollection** class.

Protected Fields

	Name	Description
	InnerList	Gets a **List<T>** containing the list of elements in the collection instance. (Inherited from CollectionBase.)
	itemEventArgs	A ItemEventArgs instance used for passing arguments to the event handlers for the events of this collection. Used to avoid creating many small objects that would burden the garbage collector. (Inherited from ItemCollectionBase.)
	notifyParent	If enabled, the collection raises events to let their containing objects handle the addition or removal of items. (Inherited from ItemCollectionBase.)

Public Properties

	Name	Description
	Count	Gets the number of elements actually contained in the collection. (Inherited from CollectionBase.)
	IsReadOnly	Gets a value indicating whether the collection is read-only. (Inherited from CollectionBase.)
	Item	Gets or sets the element at the specified index. (Inherited from CollectionBase.)

Public Methods

	Name	Description
	Add	Adds the specified diagram item to the collection. (Inherited from ItemCollectionBase.)
	Clear	Removes all diagram items from the collection. (Inherited from ItemCollectionBase.)
	Clone	Creates a shallow copy of this collection.
	Contains	Determines whether an element is in the collection. (Inherited from CollectionBase.)
	CopyTo	Copies the entire collection to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from CollectionBase.)
	GetAt	Gets the diagram item at the specified position. (Inherited from ItemCollectionBase.)

	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from CollectionBase.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire collection. (Inherited from CollectionBase.)
	Insert	Inserts the specified diagram item into the collection at the specified position. (Inherited from ItemCollectionBase.)
	Remove	Removes the diagram item from the collection. (Inherited from ItemCollectionBase.)
	RemoveAt	Removes the diagram item from the collection. (Inherited from ItemCollectionBase.)
	Sort	Overloaded. (Inherited from CollectionBase.)

Protected Methods

	Name	Description
	RaiseAdding	Raises the **Adding** event. (Inherited from ItemCollectionBase.)
	RaiseInserting	Raises the **Inserting** event. (Inherited from ItemCollectionBase.)
	RaiseRemoving	Raises the **Removing** event. (Inherited from ItemCollectionBase.)

See Also

DiagramNodeCollection Class
MindFusion.Diagramming Namespace

DiagramNodeCollection Constructor

Initializes a new instance of the DiagramNodeCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeCollection ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramNodeCollection Class
MindFusion.Diagramming Namespace

DiagramNodeCollection Methods

DiagramNodeCollection.Clone Method

Creates a shallow copy of this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeCollection Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As DiagramNodeCollection
```

Return Value

An instance of the DiagramNodeCollection class.

See Also

DiagramNodeCollection Members
DiagramNodeCollection Class
MindFusion.Diagramming Namespace

DiagramNodeEventArgs Class

Contains the arguments passed to the **NodeSelected** and **NodeRemoved** event handlers.

Syntax

C# Copy Code

```
public class DiagramNodeEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class DiagramNodeEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.DiagramNodeEventArgs

See Also

DiagramNodeEventArgs Members
MindFusion.Diagramming Namespace

DiagramNodeEventArgs Members

Contains the arguments passed to the **NodeSelected** and **NodeRemoved** event handlers.

The following tables list the members exposed by the DiagramNodeEventArgs type.

Public Constructors

	Name	Description
	DiagramNodeEventArgs	Overloaded. Initializes a new instance of the **DiagramNodeEventArgs** class.

Public Properties

	Name	Description
	Index	Gets or sets the index of the node within its containing collection.

	Node	Gets the node related to the event.

See Also

DiagramNodeEventArgs Class
MindFusion.Diagramming Namespace

DiagramNodeEventArgs Constructor

Initializes a new instance of the DiagramNodeEventArgs class.

Overload List

Name	Description
DiagramNodeEventArgs (DiagramNode)	Initializes a new instance of the **DiagramNodeEventArgs** class.
DiagramNodeEventArgs (DiagramNode, Int32)	Initializes a new instance of the **DiagramNodeEventArgs** class.

See Also

DiagramNodeEventArgs Class
MindFusion.Diagramming Namespace

DiagramNodeEventArgs Constructor (DiagramNode)

Initializes a new instance of the DiagramNodeEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeEventArgs (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode _
)
```

Parameters

node
The node related to the event.

See Also

DiagramNodeEventArgs Constructor Overload List
DiagramNodeEventArgs Class
MindFusion.Diagramming Namespace

DiagramNodeEventArgs Constructor (DiagramNode, Int32)

Initializes a new instance of the DiagramNodeEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeEventArgs (
    DiagramNode node,
    int index
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode, _
    index As Integer _
)
```

Parameters

node
The node related to the event.

index
The zero-based index of the node within its containing collection.

See Also

DiagramNodeEventArgs Constructor Overload List
DiagramNodeEventArgs Class
MindFusion.Diagramming Namespace

DiagramNodeEventArgs Properties

DiagramNodeEventArgs.Index Property

Gets or sets the index of the node within its containing collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Index { get; set; }
```

Visual Basic Copy Code

```
Public Property Index As Integer
```

Property Value

An integer value.

See Also

DiagramNodeEventArgs Members
DiagramNodeEventArgs Class
MindFusion.Diagramming Namespace

DiagramNodeEventArgs.Node Property

Gets the node related to the event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Node { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Node As DiagramNode
```

Property Value

A DiagramNode object.

See Also

DiagramNodeEventArgs Members
DiagramNodeEventArgs Class
MindFusion.Diagramming Namespace

DiagramNodeProperties Class

The base class for classes that store the properties for a DiagramNode-derived items during undo / redo operations.

Syntax

C# Copy Code

```
public class DiagramNodeProperties : DiagramItemProperties
```

Visual Basic Copy Code

```
Public Class DiagramNodeProperties
    Inherits DiagramItemProperties
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItemProperties
    MindFusion.Diagramming.DiagramNodeProperties
      MindFusion.Diagramming.ContainerNodeProperties
      MindFusion.Diagramming.ShapeNodeProperties
      MindFusion.Diagramming.TableNodeProperties

See Also

DiagramNodeProperties Members
MindFusion.Diagramming Namespace

DiagramNodeProperties Members

The base class for classes that store the properties for a DiagramNode-derived items during undo / redo operations.

The following tables list the members exposed by the DiagramNodeProperties type.

Public Constructors

	Name	Description
	DiagramNodeProperties	Initializes a new instance of the **DiagramNodeProperties** class.

Public Fields

	Name	Description

	ItemClass	For internal use. (Inherited from DiagramItemProperties.)
	labels	For internal use.
	Selected	For internal use. (Inherited from DiagramItemProperties.)
	Text	For internal use. (Inherited from DiagramItemProperties.)
	TextPadding	For internal use. (Inherited from DiagramItemProperties.)
	ZIndex	For internal use. (Inherited from DiagramItemProperties.)

Public Methods

	Name	Description
	FreeResources	Frees stored resources which will no longer be used. (Inherited from DiagramItemProperties.)

See Also

DiagramNodeProperties Class
MindFusion.Diagramming Namespace

DiagramNodeProperties Fields

DiagramNodeProperties.labels Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<NodeLabel> labels
```

Visual Basic Copy Code

```
Public labels As List(Of NodeLabel)
```

See Also

DiagramNodeProperties Members
DiagramNodeProperties Class
MindFusion.Diagramming Namespace

DiagramNodeProperties Constructor

Initializes a new instance of the DiagramNodeProperties class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeProperties ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramNodeProperties Class
MindFusion.Diagramming Namespace

DiagramNodeState Class

Contains a snapshot of the internal state of a DiagramNode.

Syntax

C# Copy Code

```
public class DiagramNodeState : DiagramItemState
```

Visual Basic Copy Code

```
Public Class DiagramNodeState
    Inherits DiagramItemState
```

Remarks

Instances of this class are used to store the state of nodes for later undo or redo operations. If you derive from DiagramNode and need to implement undo support for the additional internal state defined by your class, do so by deriving from **DiagramNodeState** and overriding the DiagramNode.CreateState, DiagramNode.SaveState and DiagramNode.RestoreState methods.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItemState
    MindFusion.Diagramming.DiagramNodeState
      MindFusion.Diagramming.ShapeNodeState

See Also

DiagramNodeState Members
MindFusion.Diagramming Namespace

DiagramNodeState Members

Contains a snapshot of the internal state of a DiagramNode.

The following tables list the members exposed by the DiagramNodeState type.

Public Constructors

	Name	Description
	DiagramNodeState	Initializes a new instance of the **DiagramNodeState** class.

See Also

DiagramNodeState Class
MindFusion.Diagramming Namespace

DiagramNodeState Constructor

Initializes a new instance of the DiagramNodeState class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeState ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramNodeState Class
MindFusion.Diagramming Namespace

DiagramNodeStyle Class

Represents a reusable set of properties that define the appearance of diagram nodes.

Syntax

C# Copy Code

```
public class DiagramNodeStyle : DiagramItemStyle
```

Visual Basic Copy Code

```
Public Class DiagramNodeStyle
    Inherits DiagramItemStyle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle
      MindFusion.Diagramming.DiagramNodeStyle
        MindFusion.Diagramming.CompositeNodeStyle
        MindFusion.Diagramming.ContainerNodeStyle
        MindFusion.Diagramming.FreeFormNodeStyle
        MindFusion.Diagramming.ShapeNodeStyle
        MindFusion.Diagramming.SvgNodeStyle
        MindFusion.Diagramming.TableNodeStyle
        MindFusion.Diagramming.TreeViewNodeStyle
        MindFusion.Diagramming.Mvc.ControlNodeStyle

See Also

DiagramNodeStyle Members
MindFusion.Diagramming Namespace

DiagramNodeStyle Members

Represents a reusable set of properties that define the appearance of diagram nodes.

The following tables list the members exposed by the DiagramNodeStyle type.

Protected Constructors

	Name	Description
	DiagramNodeStyle	Initializes a new instance of the **DiagramNodeStyle** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of the item. (Inherited from DiagramItemStyle.)
	Effects	Gets a list of the effects applied to the node.
	FontFamily	Gets or sets the font family of the item text. (Inherited from DiagramItemStyle.)
	FontSize	Gets or sets the font size of the item text. (Inherited from DiagramItemStyle.)
	FontStyle	Gets or sets the font style of the item text. (Inherited from DiagramItemStyle.)
	FontUnit	Gets or sets the font unit of the item text. (Inherited from DiagramItemStyle.)
	ShadowBrush	Gets or sets the brush used to paint the shadow of the item. (Inherited from DiagramItemStyle.)
	Stroke	Gets or sets the brush used to paint the outline of the item. (Inherited from DiagramItemStyle.)
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items. (Inherited from DiagramItemStyle.)
	StrokeThickness	Gets or sets the thickness of the item's outline, expressed in the current units of measure. (Inherited from DiagramItemStyle.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItemStyle.)

Public Methods

	Name	Description
	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
S	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JsonObject.
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)
	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a JsonObject.
	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

DiagramNodeStyle Class
MindFusion.Diagramming Namespace

DiagramNodeStyle Constructor

Initializes a new instance of the DiagramNodeStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected DiagramNodeStyle ()
```

Visual Basic Copy Code

```
Protected Sub New ()
```

See Also

DiagramNodeStyle Class
MindFusion.Diagramming Namespace

DiagramNodeStyle Methods

DiagramNodeStyle.LoadFromJson Method

Loads the style content from a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the style's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

DiagramNodeStyle Members
DiagramNodeStyle Class
MindFusion.Diagramming Namespace

DiagramNodeStyle.SaveToJson Method

Saves the style content into a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject the style's data should be stored into.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

DiagramNodeStyle Members
DiagramNodeStyle Class
MindFusion.Diagramming Namespace

DiagramNodeStyle Properties

DiagramNodeStyle.Effects Property

Gets a list of the effects applied to the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IList<NodeEffect> Effects { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Effects As IList(Of NodeEffect)
```

Property Value

A list of NodeEffect objects.

See Also

DiagramNodeStyle Members
DiagramNodeStyle Class
MindFusion.Diagramming Namespace

DiagramNodeStyle Events

DiagramPage Class

Represents a single page in a DiagramDocument object.

Syntax

C# Copy Code

```
public class DiagramPage : Diagram
```

Visual Basic Copy Code

```
Public Class DiagramPage
    Inherits Diagram
```

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.DiagramBase
        MindFusion.Diagramming.Diagram
          MindFusion.Diagramming.DiagramPage

See Also

DiagramPage Members
MindFusion.Diagramming Namespace

DiagramPage Members

Represents a single page in a DiagramDocument object.

The following tables list the members exposed by the DiagramPage type.

Public Constructors

	Name	Description
	DiagramPage	Initializes a new instance of the **DiagramPage** class.

Public Fields

	Name	Description
	Dummy	A shared DummyNode instance. (Inherited from Diagram.)

Public Properties

	Name	Description

	ActiveItem	Gets or sets the active diagram element. (Inherited from Diagram.)
	ActiveItemHandlesStyle	Gets the visual style of adjustment handles of the active item. (Inherited from Diagram.)
	ActiveLayer	Gets or sets the zero-based index of the active layer. (Inherited from Diagram.)
	AdjustmentHandlesSize	Gets or sets the size of the handles, which appear around selected objects. (Inherited from Diagram.)
	AlignmentGuidePen	Gets or sets the pen used to draw alignment guides. (Inherited from Diagram.)
	AlignToGrid	Gets or sets a value indicating whether the diagram items is aligned to a grid. (Inherited from Diagram.)
	AllowLinksRepeat	Gets or sets a value indicating whether more than one link can connect the same origin and destination diagram nodes. (Inherited from Diagram.)
	AllowMultipleResize	Gets or sets whether multiple selected nodes can be resized simultaneously. (Inherited from Diagram.)
	AllowSelfLoops	Gets or sets a value indicating whether self-loops are allowed. (Inherited from Diagram.)
	AllowSplitLinks	Gets or sets a value indicating whether link segments can be added and removed interactively. (Inherited from Diagram.)
	AllowUnanchoredLinks	Gets or sets a value indicating whether links can be connected to nodes to which there aren't anchor points assigned. (Inherited from Diagram.)
	AllowUnconnectedLinks	Gets or sets a value indicating whether users are allowed to draw links that are not connected to any node. (Inherited from Diagram.)
	AutoAlignDistance	Gets or sets the maximal horizontal or vertical distance at which a node aligns to the borders of other nodes. (Inherited from Diagram.)
	AutoAlignNodes	Gets or sets whether nodes should automatically align to the borders of other nodes during user interaction. (Inherited from Diagram.)
	AutoArrangeAvoidContainers	Gets or sets a value indicating whether auto-arranged link labels should avoid containers. (Inherited from Diagram.)
	AutoArrangeAvoidSegments	Gets or sets a value indicating whether auto-arranged link labels should avoid link segments. (Inherited from Diagram.)
	AutoCloseDistance	Gets or sets the maximum distance between first and last points of a FreeFormNode for which the node's outline is closed automatically. (Inherited from Diagram.)
	AutoDeleteLinks	Gets or sets whether links are deleted automatically when their Origin or Destination is deleted. (Inherited from Diagram.)
	AutoHighlightRows	Gets or sets whether a table row should be highlighted when the user clicks on it. (Inherited from Diagram.)
	AutoResize	Gets or sets a value indicating whether the document area must be resized automatically so it fits the diagram contents. (Inherited from Diagram.)
	AutoSnapDistance	Specifies the maximum distance at which links auto-snap to nodes. (Inherited from Diagram.)
	AutoSnapLinks	Gets or sets the default value for the AutoSnapToNode property of new links. (Inherited from Diagram.)
	BackBrush	Gets or sets the brush used for painting the diagram background. (Inherited from Diagram.)
	BackgroundImage	Gets or sets the image displayed as background of this diagram. (Inherited from Diagram.)

	BackgroundImageAlign	Gets or sets a value indicating how the background image is positioned, relatively to the scrollable document area. (Inherited from Diagram.)
	BackgroundImageUrl	URL of the image painted as diagram's background. (Inherited from Diagram.)
	BarcodeNodeStyle	Gets the theme-level style for barcode nodes. (Inherited from Diagram.)
	Bounds	Gets or sets a value representing the size of the diagram's scrollable area. (Inherited from Diagram.)
	BoundsPen	The pen with which the document bounds should be drawn. (Inherited from Diagram.)
	BranchIndicatorColor	Gets or sets the color of link branch indicators. (Inherited from Diagram.)
	BranchIndicatorSize	Gets or sets the size of link branch indicators. (Inherited from Diagram.)
	CellCustomDraw	Gets or sets the default type of custom painting done on the cells of tables. (Inherited from Diagram.)
	CellFrameStyle	Gets or sets the default cell border style of new tables. (Inherited from Diagram.)
	CompositeNodeStyle	Gets the theme-level style for composite nodes. (Inherited from Diagram.)
	ContainerCaption	Gets or sets the default value for the Caption property of containers. (Inherited from Diagram.)
	ContainerCaptionHeight	Gets or sets the default value for the CaptionHeight property of containers. (Inherited from Diagram.)
	ContainerHighlightPen	Gets or sets the default value for the HighlightPen property of containers. (Inherited from Diagram.)
	ContainerMargin	Gets or sets the default value for the Margin property of containers. (Inherited from Diagram.)
	ContainerMinimumSize	Gets or sets the default value for the MinimumSize property of containers. (Inherited from Diagram.)
	ContainerNodeStyle	Gets the theme-level style for container nodes. (Inherited from Diagram.)
	ContainersFoldable	Gets or sets the default value for the Foldable property of containers. (Inherited from Diagram.)
	CrossingRadius	Gets or sets the radius length of decorations displayed at link intersection points. (Inherited from Diagram.)
	DefaultShape	Gets or sets the default shape of shape nodes. (Inherited from Diagram.)
	DiagramLinkStyle	Gets the theme-level style for diagram links. (Inherited from Diagram.)
	DiagramStyle	Gets the theme-level style for the diagram. (Inherited from Diagram.)
	Dirty	Gets or sets a value indicating whether the diagram has changed since the last save. (Inherited from Diagram.)
	DisabledHandlesStyle	Gets the visual style of disabled adjustment handles. (Inherited from Diagram.)
	DynamicLinks	Gets or sets the default value for the Dynamic property of new links. (Inherited from Diagram.)
	EffectiveBackBrush	Gets the currently effective background brush of the diagram. (Inherited from Diagram.)
	EffectsProvider	Gets or sets an object used to generate visual effects for shape nodes such as embossment and shadow dispersion. (Inherited from Diagram.)
	EnableEmbossEffects	Gets or sets a value indicating whether emboss effect for shape nodes is enabled. (Inherited from Diagram.)

	EnableLanes	Gets or sets whether lanes are enabled. (Inherited from Diagram.)
	EnableShadowEffects	Gets or sets a value indicating whether the shadow dispersion effect for nodes is enabled. (Inherited from Diagram.)
	EnableSpatialIndex	Gets or sets whether to index item positions for faster hit-testing and viewport clipping queries. (Inherited from Diagram.)
	EnableStyledText	Gets or sets the default value for the EnableStyledText property of new nodes. (Inherited from Diagram.)
	ExpandButtonAction	Gets or sets a value indicating what should happen when a user clicks the [±] button that is displayed near expandable nodes. (Inherited from Diagram.)
	ExpandButtonPosition	Gets or sets where expand/collapse buttons are located relatively to their nodes. (Inherited from Diagram.)
	ExpandOnIncoming	Gets or sets a value, indicating in which direction object hierarchies are expanded. (Inherited from Diagram.)
	ExteriorBrush	Gets or sets the brush used for painting the area outside **Bounds**. (Inherited from Diagram.)
	Factory	Gets a Factory instance that lets you add programmatically new items to the diagram. (Inherited from Diagram.)
	FocusedComponent	Gets or sets the component within this diagram which currently has the input focus. (Inherited from Diagram.)
	FoldedContainerSize	Gets or sets the size of folded containers. (Inherited from Diagram.)
	Font	Gets or sets the default font for new diagram items. (Inherited from Diagram.)
	FreeFormAttractDistance	Gets or sets the radius around dragged free-form adjustment handle in which other points of FreeFormNode are modified too. (Inherited from Diagram.)
	FreeFormTargets	An array of Shape objects used to replace FreeFormNode instances with ShapeNode ones when Behavior is set to LinkFreeShapes or DrawFreeShapes. (Inherited from Diagram.)
	GridColor	Gets or sets the color of the grid points. (Inherited from Diagram.)
	GridOffsetX	Gets or sets the horizontal offset of the first point of the alignment grid. (Inherited from Diagram.)
	GridOffsetY	Gets or sets the vertical offset of the first point of the alignment grid. (Inherited from Diagram.)
	GridPatternHatch	Gets or sets a hatch style used to draw dense alignment grid at low zoom levels. (Inherited from Diagram.)
	GridPatternThreshold	Gets or sets the distance between grid points (in pixels) that triggers drawing using **GridPatternHatch**. (Inherited from Diagram.)
	GridSizeX	Gets or sets the horizontal distance between adjacent grid points. (Inherited from Diagram.)
	GridSizeY	Gets or sets the vertical distance between adjacent grid points. (Inherited from Diagram.)
	GridStyle	Gets or sets the visual style of the alignment grid. (Inherited from Diagram.)
	Groups	Gets a collection of all groups in the diagram. (Inherited from Diagram.)
	HitTestPriority	Gets or sets a value indicating the order in which items are hit-tested. (Inherited from Diagram.)
	Icon	Gets or sets an image associated with the page.
	IconUrl	Gets or sets the URL of the image associated with the page.

	Interaction	Gets an InteractionState instance representing the current action being carried out by the user. (Inherited from Diagram.)
	IsProEdition	Gets a value indicating whether the currently loaded MvcDiagram is a professional version of the control. (Inherited from Diagram.)
	Items	Gets a collection of all items in this diagram. (Inherited from Diagram.)
	LaneGrid	Gets the Grid instance that represents the lane grid. (Inherited from Diagram.)
	Layers	Gets a list of the layers in the current diagram. (Inherited from Diagram.)
	LinkBaseShape	Gets or sets the shape displayed at the base of new links. (Inherited from Diagram.)
	LinkBaseShapeSize	Gets or sets the default size of the base of new links. (Inherited from Diagram.)
	LinkBranchIndicator	Gets or sets a value indicating what shapes to display as markers at points where overlapping Cascading links diverge. (Inherited from Diagram.)
	LinkBrush	Gets or sets the default value for the Brush property of links. (Inherited from Diagram.)
	LinkCascadeOrientation	Gets or sets the initial orientation of the first segment of new cascading links. (Inherited from Diagram.)
	LinkCrossings	Gets or sets how to render the intersection points where links cross their paths. (Inherited from Diagram.)
	LinkCustomDraw	Gets or sets the default type of custom painting done on links. (Inherited from Diagram.)
	LinkEndsMovable	Gets or sets a value indicating whether link ends can be moved by users after the link is created. (Inherited from Diagram.)
	LinkHandlesStyle	Gets or sets the default value for the HandlesStyle property of new links. (Inherited from Diagram.)
	LinkHeadShape	Gets or sets the shape displayed as head of new links. (Inherited from Diagram.)
	LinkHeadShapeSize	Gets or sets the default size of arrowheads. (Inherited from Diagram.)
	LinkHitDistance	Gets or sets a value specifying how far from a link a click is still considered a hit. (Inherited from Diagram.)
	LinkIntermediateShape	Gets or sets the default shape to be displayed in the middle of each segment of links. (Inherited from Diagram.)
	LinkIntermediateShapeSize	Gets or sets the default size of the shapes at middles of link segments. (Inherited from Diagram.)
	LinkMergeIndicator	Gets or sets a value indicating what shapes to display as markers at points where overlapping **Cascading** links merge. (Inherited from Diagram.)
	LinkPen	Gets or sets the default value for the Pen and HeadPen properties of links. (Inherited from Diagram.)
	LinkRouter	Gets or sets the component used to find paths for auto-routed links when diagram nodes are added or their positions change. (Inherited from Diagram.)
	Links	Gets a collection of all DiagramLink objects in the diagram. (Inherited from Diagram.)
	LinkSegments	Gets or sets the default number of link segments. (Inherited from Diagram.)
	LinkShape	Gets or sets the default value of the Shape property of new links. (Inherited from Diagram.)

	LinksRetainForm	Gets or sets whether links should maintain their forms if the nodes linked with those links are moved around. (Inherited from Diagram.)
	LinkText	Gets or sets the default text of links. (Inherited from Diagram.)
	LinkTextStyle	Gets or sets default placement and orientation for links' text. (Inherited from Diagram.)
	MeasureTextRenderingHint	Gets or sets the TextRenderingHint to use when measuring texts. This should be effectively set to the value of DiagramView.TextRenderingHint. (Inherited from Diagram.)
	MeasureUnit	Gets or sets the unit of measure used for logical coordinates in this diagram. (Inherited from Diagram.)
	MergeIndicatorColor	Gets or sets the color of link merge indicators. (Inherited from Diagram.)
	MergeIndicatorSize	Gets or sets the size of link merge indicators. (Inherited from Diagram.)
	MergeThreshold	Gets or sets the maximum distance between adjacent control points of a link at which the respective segments can be merged. (Inherited from Diagram.)
	MinimumNodeSize	Gets or sets the minimum node size that the user is allowed to set. (Inherited from Diagram.)
	ModificationEffect	Gets or sets the visual effect to apply during interactive modification of items. (Inherited from Diagram.)
	NodeEffects	Gets a collection with all effects applied to the nodes of this diagram. (Inherited from Diagram.)
	Nodes	Gets a collection of all DiagramNode objects in the diagram. (Inherited from Diagram.)
	NodesExpandable	Gets or sets a value indicating whether newly created nodes can be expanded. (Inherited from Diagram.)
	NowLoading	Gets a value indicating whether the diagram data is now being loaded from a file. (Inherited from Diagram.)
	PolygonalTextLayout	Gets or sets a value indicating whether to enable the formatting of text inside non-rectangular shape node outlines. (Inherited from Diagram.)
	RecursiveExpand	Gets or sets value indicating how a tree is expanded. (Inherited from Diagram.)
	RenderOptions	Gets the current rendering options for this diagram. (Inherited from Diagram.)
	RestrictItemsToBounds	Gets or sets a value indicating whether users are allowed to place items outside the scrollable area of the diagram. (Inherited from Diagram.)
	RoundedLinks	Gets or sets a value indicating whether the joins between segments of **Polyline** and **Cascading** links should be rounded. (Inherited from Diagram.)
	RoundedLinksRadius	Gets or sets the radius of arc joins of rounded links' segments. (Inherited from Diagram.)
	RoundRectFactor	Gets or sets the relative size of arcs displayed at the corners of rounded rectangles. (Inherited from Diagram.)
	RouteLinks	Gets or sets a value indicating whether links should avoid nodes by going the shortest path from their origin to their destination without crossing any other nodes. (Inherited from Diagram.)
	RoutingOptions	Gets an object, which allows customizing various parameters of the link routing algorithm. (Inherited from Diagram.)
	RowHighlightBrush	Gets or sets the Brush used to fill highlighted table rows. (Inherited from Diagram.)
	ScriptHelper	Gets a ScriptHelper object which can be used in html scripts to instantiate various objects and to call static methods. (Inherited from

		Diagram.)
	SelectAfterCreate	Gets or sets a value indicating whether newly created objects are automatically selected. (Inherited from Diagram.)
	SelectedItemHandlesStyle	Gets the visual style of adjustment handles of selected items. (Inherited from Diagram.)
	Selection	Gets a Selection instance that represents the selection of items in this diagram. (Inherited from Diagram.)
	SelectionOnTop	Gets or sets a value indicating whether selected items should be painted on top of other items. (Inherited from Diagram.)
	SerializationBinder	Gets or sets a SerializationBinder object used to control class loading. (Inherited from Diagram.)
	ShadowBrush	Gets or sets the default brush used to paint the shadows cast by diagram items. (Inherited from Diagram.)
	ShadowOffsetX	Gets or sets the horizontal offset of item shadows. (Inherited from Diagram.)
	ShadowOffsetY	Gets or sets the vertical offset of item shadows. (Inherited from Diagram.)
	ShadowsStyle	Gets or sets the style for rendering shadows. (Inherited from Diagram.)
	ShapeBrush	Gets or sets the default value for the **Brush** property of shape nodes. (Inherited from Diagram.)
	ShapeCustomDraw	Gets or sets the default type of custom painting done for shape nodes. (Inherited from Diagram.)
	ShapeHandlesStyle	Gets or sets the default value for the **HandlesStyle** property of new shape nodes. (Inherited from Diagram.)
	ShapeNodeStyle	Gets the theme-level style for shape nodes. (Inherited from Diagram.)
	ShapeOrientation	Gets or sets the default value for the ShapeOrientation property of shape nodes. (Inherited from Diagram.)
	ShapePen	Gets or sets the default value for the **Pen** property of shape nodes. (Inherited from Diagram.)
	ShapeText	Gets or sets the default text of shape nodes. (Inherited from Diagram.)
	ShowAnchors	Gets or sets a value indicating whether and when node anchor points should be displayed. (Inherited from Diagram.)
	ShowDisabledHandles	Gets or sets a value indicating whether manipulation handles are painted when they are disabled. (Inherited from Diagram.)
	ShowGrid	Gets or sets a value indicating whether the alignment grid is visible. (Inherited from Diagram.)
	ShowHandlesOnDrag	Gets or sets a value indicating whether to display selection handles of objects under the mouse while another object is being modified. (Inherited from Diagram.)
	SnapToAnchor	Gets or sets when link end points should be aligned to the nearest anchor points. (Inherited from Diagram.)
	SortGroupsByZ	Gets or sets a value indicating whether grouped items should be drawn sorted by their z-order while the whole group is being moved. (Inherited from Diagram.)
	Style	Gets or sets the style associated with the diagram. (Inherited from Diagram.)
	SvgNodeStyle	Gets the theme-level style for SVG nodes. (Inherited from Diagram.)
	TableBrush	Gets or sets the default value for the **Brush** property of tables. (Inherited from Diagram.)

	TableCaption	Gets or sets the default text for tables' captions. (Inherited from Diagram.)
	TableCaptionHeight	Gets or sets the default height of table captions. (Inherited from Diagram.)
	TableColumnCount	Gets or sets the default number of table columns. (Inherited from Diagram.)
	TableColumnWidth	Gets or sets the default width of tables' columns. (Inherited from Diagram.)
	TableConnectionStyle	Gets or sets a value indicating whether drawing a link between tables should connect the rows of those tables, or the tables as whole entities. (Inherited from Diagram.)
	TableCustomDraw	Gets or sets the default type of custom painting done on tables. (Inherited from Diagram.)
	TableHandlesStyle	Gets or sets the default value for the **HandlesStyle** property of new tables. (Inherited from Diagram.)
	TableNodeStyle	Gets the theme-level style for table nodes. (Inherited from Diagram.)
	TablePen	Gets or sets the default value for the **Pen** property of tables. (Inherited from Diagram.)
	TableRowCount	Gets or sets the default number of table rows. (Inherited from Diagram.)
	TableRowHeight	Gets or sets the default height of tables' rows. (Inherited from Diagram.)
	TableShape	Gets or sets the default value of the Shape property of new tables. (Inherited from Diagram.)
	TablesScrollable	Gets or sets the default value for the Scrollable property of new tables. (Inherited from Diagram.)
	Tag	Gets or sets custom data associated with this diagram. (Inherited from Diagram.)
	TextColor	Gets or sets the default color of text displayed inside diagram items. (Inherited from Diagram.)
	TextFormat	Gets or sets the default alignment and formatting style for text displayed inside nodes and table cells. (Inherited from Diagram.)
	Theme	Gets or sets the theme associated with the diagram. (Inherited from Diagram.)
	Title	Gets or sets the page title.
	TitleColor	Gets or sets the color of the title when the title is displayed.
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram. (Inherited from Diagram.)
	TouchHitDistance	Specifies distance from adjustment handles at which they are still considered as hit when processing touch input. (Inherited from Diagram.)
	TouchThreshold	Gets or sets the threshold value for easier touch interaction through adjustment handles. (Inherited from Diagram.)
	TreeViewNodeStyle	Gets the theme-level style for tree view nodes. (Inherited from Diagram.)
	UndoManager	Gets a reference to the diagram's UndoManager instance. (Inherited from Diagram.)
	ValidityChecks	Gets or sets a value indicating whether validity checks are performed each time an item is added to the diagram. (Inherited from Diagram.)
	VisioNodeStyle	Gets the theme-level style for Visio nodes. (Inherited from Diagram.)
	XmlOptions	An XmlOptions object providing properties for customizing the XML serialization process. (Inherited from DiagramBase.)

Public Methods

	Name	Description
	AlignPointToGrid	Returns a point of the alignment grid nearest to the one passed as an argument. (Inherited from Diagram.)
	AllowNodeBounds	Determines whether node can be placed at specified coordinates. (Inherited from Diagram.)
	Arrange	Arranges the diagram using the specified graph layout algorithm. (Inherited from Diagram.)
	ArrangeAnimated	Arranges the diagram using the specified graph layout algorithm, and animates items moving to their new positions. (Inherited from Diagram.)
	ArrangeAsync	Arranges the diagram asynchronously using the specified graph layout algorithm. (Inherited from Diagram.)
	ArrangeLinkLabels	Rearranges link labels whose AutoArrange property is enabled. (Inherited from Diagram.)
	ChangeUnit	Sets new **MeasureUnit** value and scales the coordinates of diagram items so that their absolute size remains unchanged. (Inherited from Diagram.)
	ClearAll	Removes all items from the diagram. (Inherited from Diagram.)
	Clone	Creates a copy of this diagram page.
	CopySelection	Overloaded. Creates a copy of specified diagram elements. (Inherited from Diagram.)
	CopyTo	Copies this page's properties and items to specified target diagram.
	CreateImage	Overloaded. Creates a bitmap containing an image of the diagram. (Inherited from Diagram.)
	CreateMeasureGraphics	Creates an IGraphics object that can be used to obtain various string size measurements. (Inherited from Diagram.)
	CreateThemeFromDefaults	Creates a Theme object representing the default properties of the diagram. (Inherited from Diagram.)
	Draw	Renders the diagram on the specified **IGraphics** object. (Inherited from Diagram.)
	DrawAlignmentGuides	Draws the current alignment guides. (Inherited from Diagram.)
	DrawInteraction	Draws the items that the user is currently modifying. (Inherited from Diagram.)
	DrawInteractions	For internal use. (Inherited from Diagram.)
	DrawLayer	Renders all diagram items to the specified graphics. (Inherited from Diagram.)
S	DrawStyledText	Draws a formatted text string, the same way text is drawn in shape nodes whose **EnableStyledText** property is set to **true**. This method lets you display styled text in custom-drawn nodes or links. (Inherited from Diagram.)
	EffectiveTouchHitDistance	Returns **TouchHitDistance** or a default value. (Inherited from Diagram.)
	EnumByReverseZ	Hit-tests diagram items in reverse Z order. (Inherited from Diagram.)
	EnumTopLevelByReverseZ	Hit-tests diagram top-level items in reverse Z order. (Inherited from Diagram.)
	ExecuteCommand	Executes the specified command on this diagram. (Inherited from Diagram.)
	FindGroup	Finds the group that has the specified tag value. (Inherited from Diagram.)

	FindLink	Finds the link that has the specified tag value. (Inherited from Diagram.)
	FindLinkById	Finds the link that has the specified id value. (Inherited from Diagram.)
	FindNode	Finds the node that has the specified tag value. (Inherited from Diagram.)
	FindNodeById	Finds the node that has the specified id value. (Inherited from Diagram.)
	GetContentBounds	Returns the smallest rectangle that bounds all diagram objects. (Inherited from Diagram.)
	GetItemAt	Overloaded. (Inherited from Diagram.)
	GetLinkAt	Overloaded. Finds and returns the link that lies at the specified location. (Inherited from Diagram.)
	GetLinksAt	Returns the links that contain the specified point. (Inherited from Diagram.)
	GetNearestNode	Returns the node nearest to the specified point. (Inherited from Diagram.)
	GetNodeAt	Overloaded. Gets the node that lies at the specified location. (Inherited from Diagram.)
	GetNodesAt	Returns the nodes that contain the specified point. (Inherited from Diagram.)
	GetNodesInViewport	Returns all nodes visible in the current viewport. (Inherited from Diagram.)
	HandleKeyDown	Propagates keyboard input to interested manipulators and components. (Inherited from Diagram.)
	HandleKeyPress	Propagates keyboard input to interested manipulators. (Inherited from Diagram.)
	HandleKeyUp	Propagates keyboard input to interested components. (Inherited from Diagram.)
	HandleMouseMove	Propagates mouse input to interested manipulators. (Inherited from Diagram.)
	HitTestComponents	Tests whether the specified point is within a component of a diagram item and returns this component. (Inherited from Diagram.)
	HitTestLabels	Returns the item label located at specified position. (Inherited from Diagram.)
	HitTestManipulators	Tests whether the specified point is within the manipulator of a diagram object and triggers the manipulator. (Inherited from Diagram.)
	Invalidate	Overloaded. Invalidates the diagram or a region of the diagram, causing it to be repainted. (Inherited from Diagram.)
	InvalidateZOrder	Forces a new sort of the Z order list before next redraw. (Inherited from Diagram.)
	IsClassRegistered	Checks if a class is registered for serialization support. (Inherited from Diagram.)
	IsClientClassRegistered	Checks if a class is registered for JSON serialization support. (Inherited from Diagram.)
	IsItemInteractive	Checks if the specified item can be modified interactively. (Inherited from Diagram.)
	IsItemLocked	Checks if the specified item is locked. (Inherited from Diagram.)
	IsItemVisible	Checks if the specified item is currently visible. (Inherited from Diagram.)
	LoadFromJson	Loads the diagram from a JSON-formatted string. (Inherited from Diagram.)

	LoadFromJsonAsync	Loads the diagram from a JSON-formatted string. (Inherited from Diagram.)
	LoadFromJsonFile	Loads the diagram from a JSON file. (Inherited from Diagram.)
	LoadFromJsonFileAsync	Loads the diagram from a JSON file. (Inherited from Diagram.)
	LoadFromJsonObject	Loads the diagram from a JSON object. (Inherited from Diagram.)
	LoadFromString	Decodes diagram contents from a string that was created with SaveToString. (Inherited from Diagram.)
	LoadFromXml	Overloaded. Loads the contents of an XML diagram document into the control. (Inherited from Diagram.)
	LoadFromXmlAsync	Overloaded. (Inherited from Diagram.)
	LoadSelectionFromJson	(Inherited from Diagram.)
	MeasureString	Overloaded. Measures a text taking into consideration the current affine transform of the diagram. (Inherited from Diagram.)
	Navigate	Selects the nearest item in the specified direction. (Inherited from Diagram.)
	NewInstance	Custom Diagram classes must override this method to return an instance of their type. (Inherited from Diagram.)
	PasteSelection	Adds the specified items and groups to the specified diagram document. (Inherited from Diagram.)
	RaiseCellTextEdited	Raises the CellTextEdited event . (Inherited from Diagram.)
	RaiseCellTextEditing	Raises the CellTextEditing event to validate entering in-place editing mode. (Inherited from Diagram.)
	RaiseClicked	Overloaded. Raises the Clicked event of the diagram or one of the *Clicked events related to diagram items. (Inherited from Diagram.)
	RaiseCreated	Raises the appropriate *Created event according to the type of the specified item. (Inherited from Diagram.)
	RaiseDeleting	Raises the appropriate *Deleting event according to the type of the specified item. (Inherited from Diagram.)
	RaiseDoubleClicked	Overloaded. Raises the DoubleClicked event of the diagram or one of the *DoubleClicked events related to diagram items. (Inherited from Diagram.)
	RaiseLinkTextEdited	Raises the LinkTextEdited event. (Inherited from Diagram.)
	RaiseLinkTextEditing	Raises the LinkTextEditing event to validate entering inplace edit mode. (Inherited from Diagram.)
	RaiseModified	Raises the NodeModified event. (Inherited from Diagram.)
	RaiseMouseHover	Raises the appropriate Pointed event. (Inherited from Diagram.)
	RaiseNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	RaiseNodeTextEdited	Raises the NodeTextEdited event. (Inherited from Diagram.)
	RaiseNodeTextEditing	Raises the NodeTextEditing event to validate entering inplace edit mode. (Inherited from Diagram.)
	RaisePasted	Raises the appropriate Pasted event. (Inherited from Diagram.)
	RaiseSelecting	Raises the appropriate Selecting event. (Inherited from Diagram.)
	RaiseTreeItemTextEditing	Raises the TreeItemTextEditing event to validate entering inplace edit mode. (Inherited from Diagram.)
	RaiseViewRemoved	Raises the ViewRemoved event. (Inherited from Diagram.)

	RegisterClass	Overloaded. (Inherited from Diagram.)
	RegisterDefaultStyle	Registers a default style for the objects of the specified type. (Inherited from Diagram.)
	RegisterItemClass	Overloaded. (Inherited from Diagram.)
	ResetAlignmentGuides	For internal use. (Inherited from Diagram.)
	ResizeToFitItem	Resizes the document extents so that the specified item is contained within. (Inherited from Diagram.)
	ResizeToFitItems	Overloaded. Resizes the document scrollable area so it fits all diagram items. (Inherited from Diagram.)
	ResumeLinkRouter	Called when the control ends a batch update operation that might affect many links. (Inherited from Diagram.)
	RouteAllLinks	Overloaded. (Inherited from Diagram.)
	SaveSelectionToJson	Saves the diagram selection to a JSON object. (Inherited from Diagram.)
	SaveToJson	Saves the diagram to a JSON-formatted string. (Inherited from Diagram.)
	SaveToJsonAsync	Saves the diagram to a JSON-formatted string. (Inherited from Diagram.)
	SaveToJsonFile	Saves the diagram into a JSON file. (Inherited from Diagram.)
	SaveToJsonFileAsync	Saves the diagram into a JSON file. (Inherited from Diagram.)
	SaveToJsonObject	Saves the diagram to a JSON object. (Inherited from Diagram.)
	SaveToString	Encodes the diagram contents into an ASCII string using the specified format and optionally clearing the **Dirty** flag. (Inherited from Diagram.)
	SaveToXml	Overloaded. Saves the diagram to an XML document. (Inherited from Diagram.)
	SaveToXmlAsync	Overloaded. (Inherited from Diagram.)
	SetShadowOffset	Sets the default offset at which items shadows are drawn. (Inherited from Diagram.)
	SuspendLinkRouter	Called when the control starts a batch update operation that might affect many links. The diagram collects links specified through Route and RouteAllLinks calls and defers finding their paths to when **ResumeLinkRouter** is called. (Inherited from Diagram.)
	TrackMouseHover	Specifies whether the diagram view should track mouse hover events. (Inherited from DiagramBase.)
	UpdateMeasureTextScale	Gets or sets the scale to use when measuring texts. This should be effectively set to the value of DiagramView.Zoom / 100. (Inherited from Diagram.)
	UpdateRuntimeIndices	For internal use. (Inherited from Diagram.)
	UpdateZOrder	Immediately sorts the Z-order list if Z-order has been invalidated. Otherwise it will be sorted next time the diagram draws its contents. (Inherited from Diagram.)

Protected Methods

	Name	Description
	AddItem	Invoked by the AddItemCmd and RemoveItemCmd classes to effectively add the specified item to the diagram. (Inherited from Diagram.)
	CreateAddItemCmd	Creates an **AddItemCmd** instance that implements undo/redo of adding an item to the diagram. (Inherited from Diagram.)
	CreateModifyItemCmd	Creates a ModifyItemCmd instance that implements undo/redo of item move and resize operations. (Inherited from Diagram.)

	CreateRemoveItemCmd	Creates a **RemoveItemCmd** instance that implements undo/redo of removing an item from the diagram. (Inherited from Diagram.)
	DeleteGroup	Removes the specified group from the group collection. (Inherited from Diagram.)
	Dispose	Clean up any resources being used. (Inherited from Diagram.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from Diagram.)
	OnActionRecorded	Raises the ActionRecorded event. (Inherited from DiagramBase.)
	OnActionRecording	Raises the ActionRecording event. (Inherited from DiagramBase.)
	OnActionRedone	Raises the ActionRedone event. (Inherited from DiagramBase.)
	OnActionUndone	Raises the ActionUndone event. (Inherited from DiagramBase.)
	OnBeginLoad	Raises the BeginLoad event. (Inherited from DiagramBase.)
	OnBoundsChanged	Raises the BoundsChanged event. (Inherited from DiagramBase.)
	OnCellClicked	Raises the CellClicked event. (Inherited from DiagramBase.)
	OnCellDoubleClicked	Raises the CellDoubleClicked event. (Inherited from DiagramBase.)
	OnCellTextEdited	Raises the **CellTextEdited** event. (Inherited from DiagramBase.)
	OnCellTextEditing	Raises the **CellTextEditing** event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnClicked	Raises the **Clicked** event. (Inherited from DiagramBase.)
	OnContainerChildAdded	Raises the ContainerChildAdded event. (Inherited from DiagramBase.)
	OnContainerChildRemoved	Raises the ContainerChildRemoved event. (Inherited from DiagramBase.)
	OnContainerFolded	Raises the ContainerFolded event. (Inherited from DiagramBase.)
	OnContainerPageChanged	Raises the ContainerPageChanged event. (Inherited from DiagramBase.)
	OnContainerUnfolded	Raises the ContainerUnfolded event. (Inherited from DiagramBase.)
	OnContentVisibilityChanged	Raises the ContentVisibilityChanged event. (Inherited from DiagramBase.)
	OnDefaultShapeChanged	Raises the DefaultShapeChanged event. (Inherited from DiagramBase.)
	OnDeserializeTag	Raises the DeserializeTag event. (Inherited from DiagramBase.)
	OnDestinationAnchorChanged	Raises the DestinationAnchorChanged event. (Inherited from DiagramBase.)
	OnDirtyChanged	Raises the DirtyChanged event. (Inherited from DiagramBase.)
	OnDoubleClicked	Raises the **DoubleClicked** event. (Inherited from DiagramBase.)
	OnDrawAdjustmentHandles	Raises the DrawAdjustmentHandles event. (Inherited from DiagramBase.)
	OnDrawAnchorPoint	Raises the DrawAnchorPoint event. (Inherited from DiagramBase.)
	OnDrawBackground	Raises the DrawBackground event. (Inherited from DiagramBase.)
	OnDrawBranchIndicator	Raises the DrawBranchIndicator event. (Inherited from DiagramBase.)
	OnDrawCell	Raises the DrawCell event. (Inherited from DiagramBase.)
	OnDrawForeground	Raises the DrawForeground event. (Inherited from DiagramBase.)
	OnDrawLink	Raises the DrawLink event. (Inherited from DiagramBase.)
	OnDrawNode	Raises the DrawNode event. (Inherited from DiagramBase.)

	OnEndLoad	Raises the EndLoad event. (Inherited from DiagramBase.)
	OnExpandButtonClicked	Raises the ExpandButtonClicked event. (Inherited from DiagramBase.)
	OnGroupDestroyed	Raises the GroupDestroyed event. (Inherited from DiagramBase.)
	OnHitTestAdjustmentHandles	Raises the HitTestAdjustmentHandles event. (Inherited from DiagramBase.)
	OnInitializeLasso	Raises the InitializeLasso event. (Inherited from DiagramBase.)
	OnInitializeLink	Raises the InitializeLink event. (Inherited from DiagramBase.)
	OnInitializeNode	Raises the InitializeNode event. (Inherited from DiagramBase.)
	OnItemAdded	Raises the ItemAdded event. (Inherited from DiagramBase.)
	OnItemRemoved	Raises the ItemRemoved event. (Inherited from DiagramBase.)
	OnJsonDeserializeTag	Raises the JsonDeserializeTag event. (Inherited from DiagramBase.)
	OnJsonSerializeTag	Raises the JsonSerializeTag event. (Inherited from DiagramBase.)
	OnLayerVisibilityChanged	Raises the LayerVisibilityChanged event. (Inherited from DiagramBase.)
	OnLinkActivated	Raises the LinkActivated event. (Inherited from DiagramBase.)
	OnLinkClicked	Raises the LinkClicked event. (Inherited from DiagramBase.)
	OnLinkCreateCancelled	Raises the LinkCreateCancelled event. (Inherited from DiagramBase.)
	OnLinkCreated	Raises the LinkCreated event. (Inherited from DiagramBase.)
	OnLinkCreating	Raises the LinkCreating event. (Inherited from DiagramBase.)
	OnLinkDeactivated	Raises the LinkDeactivated event. (Inherited from DiagramBase.)
	OnLinkDeleted	Raises the LinkDeleted event. (Inherited from DiagramBase.)
	OnLinkDeleting	Raises the LinkDeleting event. (Inherited from DiagramBase.)
	OnLinkDeselected	Raises the LinkDeselected event. (Inherited from DiagramBase.)
	OnLinkDoubleClicked	Raises the LinkDoubleClicked event. (Inherited from DiagramBase.)
	OnLinkHovered	Raises the LinkHovered event. (Inherited from DiagramBase.)
	OnLinkModified	Raises the LinkModified event. (Inherited from DiagramBase.)
	OnLinkModifyCancelled	Raises the LinkModifyCancelled event. (Inherited from DiagramBase.)
	OnLinkModifying	Raises the LinkModifying event. (Inherited from DiagramBase.)
	OnLinkPasted	Raises the LinkPasted event. (Inherited from DiagramBase.)
	OnLinkRouted	Raises the LinkRouted event. (Inherited from DiagramBase.)
	OnLinkSelected	Raises the LinkSelected event. (Inherited from DiagramBase.)
	OnLinkSelecting	Raises the LinkSelecting event. (Inherited from DiagramBase.)
	OnLinkSplit	Raises the LinkSplit event. (Inherited from DiagramBase.)
	OnLinkStartModifying	Raises the LinkStartModifying event. (Inherited from DiagramBase.)
	OnLinkTextEdited	Raises the **LinkTextEdited** event. (Inherited from DiagramBase.)
	OnLinkTextEditing	Raises the **LinkTextEditing** event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnMeasureUnitChanged	Raises the MeasureUnitChanged event. (Inherited from DiagramBase.)
	OnNodeActivated	Raises the NodeActivated event. (Inherited from DiagramBase.)

	OnNodeClicked	Raises the NodeClicked event. (Inherited from DiagramBase.)
	OnNodeCreateCancelled	Raises the NodeCreateCancelled event. (Inherited from DiagramBase.)
	OnNodeCreated	Raises the NodeCreated event. (Inherited from DiagramBase.)
	OnNodeCreating	Raises the NodeCreating event. (Inherited from DiagramBase.)
	OnNodeDeactivated	Raises the NodeDeactivated event. (Inherited from DiagramBase.)
	OnNodeDeleted	Raises the NodeDeleted event. (Inherited from DiagramBase.)
	OnNodeDeleting	Raises the NodeDeleting event. (Inherited from DiagramBase.)
	OnNodeDeselected	Raises the NodeDeselected event. (Inherited from DiagramBase.)
	OnNodeDoubleClicked	Raises the NodeDoubleClicked event. (Inherited from DiagramBase.)
	OnNodeHovered	Raises the NodeHovered event. (Inherited from DiagramBase.)
	OnNodeModified	Raises the **NodeModified** event. (Inherited from DiagramBase.)
	OnNodeModifyCancelled	Raises the NodeModifyCancelled event. (Inherited from DiagramBase.)
	OnNodeModifying	Raises the NodeModifying event. (Inherited from DiagramBase.)
	OnNodePasted	Raises the NodePasted event. (Inherited from DiagramBase.)
	OnNodeSelected	Raises the NodeSelected event. (Inherited from DiagramBase.)
	OnNodeSelecting	Raises the NodeSelecting event. (Inherited from DiagramBase.)
	OnNodeStartModifying	Raises the NodeStartModifying event. (Inherited from DiagramBase.)
	OnNodeTextEdited	Raises the **NodeTextEdited** event. (Inherited from DiagramBase.)
	OnNodeTextEditing	Raises the NodeTextEditing event to validate entering inplace edit mode. (Inherited from DiagramBase.)
	OnOriginAnchorChanged	Raises the OriginAnchorChanged event. (Inherited from DiagramBase.)
	OnPropertyChanged	Raises the PropertyChanged event.
	OnQueryAlignTarget	Raises the QueryAlignTarget event. (Inherited from DiagramBase.)
	OnRepaint	Raises the Repaint event. (Inherited from DiagramBase.)
	OnRequestClipboard	Raises the RequestClipboard event. (Inherited from DiagramBase.)
	OnRequestDrawInteractions	Raises the RequestDrawInteractions event. (Inherited from DiagramBase.)
	OnRequestTimer	Raises the RequestTimer event. (Inherited from DiagramBase.)
	OnSelectionChanged	Raises the SelectionChanged event. (Inherited from DiagramBase.)
	OnSelectionModified	Raises the SelectionModified event. (Inherited from DiagramBase.)
	OnSelectionModifying	Raises the SelectionModifying event. (Inherited from DiagramBase.)
	OnSerializeTag	Raises the SerializeTag event. (Inherited from DiagramBase.)
	OnSetSelfLoopShape	Raises the SetSelfLoopShape event. (Inherited from DiagramBase.)
	OnShapeModified	Raises the ShapeModified event. (Inherited from DiagramBase.)
	OnShapeNotDefined	Raises the ShapeNotDefined event. (Inherited from DiagramBase.)
	OnTableColumnResized	Raises the TableColumnResized event. (Inherited from DiagramBase.)
	OnTableColumnResizing	Raises the TableColumnResizing event. (Inherited from DiagramBase.)
	OnTableRowResized	Raises the TableRowResized event. (Inherited from DiagramBase.)

	OnTableRowResizing	Raises the TableRowResizing event. (Inherited from DiagramBase.)
	OnTableSectionCollapsed	Raises the TableSectionCollapsed event. (Inherited from DiagramBase.)
	OnTableSectionExpanded	Raises the TableSectionExpanded event. (Inherited from DiagramBase.)
	OnTreeCollapsed	Raises the TreeCollapsed event. (Inherited from DiagramBase.)
	OnTreeExpanded	Raises the TreeExpanded event. (Inherited from DiagramBase.)
	OnTreeItemTextEditing	Raises the **TreeItemTextEditing** event. (Inherited from DiagramBase.)
	OnValidateAnchorPoint	Raises the ValidateAnchorPoint event. (Inherited from DiagramBase.)
	OnValidateLinkCrossing	Raises the ValidateLinkCrossing event. (Inherited from DiagramBase.)
	OnViewRemoved	Raises the **ViewRemoved** event. (Inherited from DiagramBase.)
	RemoveItem	Invoked by the **AddItemCmd** and **RemoveItemCmd** classes to effectively remove the specified item from the diagram. (Inherited from Diagram.)

Public Events

	Name	Description
	ActionRecorded	Raised when an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRecording	A validation event raised before an action is recorded in the undo buffer. (Inherited from DiagramBase.)
	ActionRedone	Raised when an action is redone. (Inherited from DiagramBase.)
	ActionUndone	Raised when an action is undone. (Inherited from DiagramBase.)
	ActiveLayerChanged	Raised when the diagram's active layer has changed. (Inherited from DiagramBase.)
	BeginLoad	Raised when the component starts loading diagram data from a file or memory stream. (Inherited from DiagramBase.)
	BoundsChanged	Raised when the **Bounds** property value has changed. (Inherited from DiagramBase.)
	CellClicked	Raised when the user clicks a table's cell. (Inherited from DiagramBase.)
	CellDoubleClicked	Raised when the user double-clicks a table's cell. (Inherited from DiagramBase.)
	CellTextEdited	Raised when the user edits the text of a table's cell. (Inherited from DiagramBase.)
	CellTextEditing	A validation event that lets you approve or reject changes made to the text of a table's cell. (Inherited from DiagramBase.)
	Clicked	Raised when the user clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	ContainerChildAdded	Raised when a node is added to a ContainerNode, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerChildRemoved	Raised when a node is removed from a **ContainerNode**, either programmatically or by the user. (Inherited from DiagramBase.)
	ContainerFolded	Raised when the user folds a container by clicking the arrow-up button in the container's caption bar. (Inherited from DiagramBase.)
	ContainerPageChanged	Raised when the user changes current page of a PagedContainerNode. (Inherited from DiagramBase.)
	ContainerUnfolded	Raised when the user unfolds a container by clicking the arrow-down button in the container's caption bar. (Inherited from DiagramBase.)

	ContentVisibilityChanged	Raised when the visibility of container's content changes. (Inherited from DiagramBase.)
	DefaultShapeChanged	Raised when the **DefaultShape** property value has changed. (Inherited from DiagramBase.)
	DeserializeTag	Raised when the Tag of a diagram item must be deserialized from XML format. (Inherited from DiagramBase.)
	DestinationAnchorChanged	Raised when the DestinationAnchor property value of a link has changed. (Inherited from DiagramBase.)
	DirtyChanged	Raised when the **Dirty** property value has changed. (Inherited from DiagramBase.)
	DoubleClicked	Raised when the user double-clicks the diagram at a point where no items are located. (Inherited from DiagramBase.)
	DrawAdjustmentHandles	Raised when item's adjustment handles must be custom-drawn. (Inherited from DiagramBase.)
	DrawAnchorPoint	Raised when an anchor point must be custom-drawn. (Inherited from DiagramBase.)
	DrawBackground	Raised to let you custom draw the diagram background. (Inherited from DiagramBase.)
	DrawBranchIndicator	Raised when a branch indicator must be custom-drawn. (Inherited from DiagramBase.)
	DrawCell	Raised when a table's cell must be custom drawn. (Inherited from DiagramBase.)
	DrawForeground	Raised to let you custom draw the diagram foreground. (Inherited from DiagramBase.)
	DrawLink	Raised when a diagram link must be custom drawn. (Inherited from DiagramBase.)
	DrawNode	Raised when a diagram node must be custom drawn. (Inherited from DiagramBase.)
	EndLoad	Raised when the component ends loading diagram data from a file or memory stream. (Inherited from DiagramBase.)
	ExpandButtonClicked	Raised when the user clicks the [±] button of an Expandable node. (Inherited from DiagramBase.)
	GroupDestroyed	Raised when a Group object is destroyed. (Inherited from DiagramBase.)
	HitTestAdjustmentHandles	Raised when custom hit testing of item's adjustment handles must be performed. (Inherited from DiagramBase.)
	InitializeLasso	Raised when the user starts drawing a selection lasso, before it is rendered on screen. (Inherited from DiagramBase.)
	InitializeLink	Raised when the user starts drawing a new link, just after the **DiagramLink** instance is created. (Inherited from DiagramBase.)
	InitializeNode	Raised when the user starts drawing a new node, just after the **DiagramNode** instance is created. (Inherited from DiagramBase.)
	ItemAdded	Raised when an item is added to the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	ItemRemoved	Raised when an item is removed from the diagram, either programmatically or interactively. (Inherited from DiagramBase.)
	JsonDeserializeTag	Raised when the **Tag** of a diagram item must be deserialized from JSON. (Inherited from DiagramBase.)
	JsonSerializeTag	Raised when the **Tag** of a diagram item must be serialized to JSON. (Inherited from DiagramBase.)

	LayerVisibilityChanged	Raised when the visibility of a Layer changes. (Inherited from DiagramBase.)
	LinkActivated	Raised when a link becomes the **ActiveItem**. (Inherited from DiagramBase.)
	LinkClicked	Raised when the user clicks a link. (Inherited from DiagramBase.)
	LinkCreateCancelled	Raised when the user cancels link creation. (Inherited from DiagramBase.)
	LinkCreated	Raised when the user draws a new link. (Inherited from DiagramBase.)
	LinkCreating	A validation event raised while the user is drawing a new link. (Inherited from DiagramBase.)
	LinkDeactivated	Raised when a link is no longer the **ActiveItem**. (Inherited from DiagramBase.)
	LinkDeleted	Raised when a link is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDeleting	Raised when the user tries to delete a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkDeselected	Raised when a link is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkDoubleClicked	Raised when the user double-clicks a link. (Inherited from DiagramBase.)
	LinkHovered	Raised when the mouse pointer hovers over a link. (Inherited from DiagramBase.)
	LinkModified	Raised when the user moves any control point of a link. (Inherited from DiagramBase.)
	LinkModifyCancelled	Raised when the user cancels move or resize operation being done on a link. (Inherited from DiagramBase.)
	LinkModifying	A validation event raised while the user is modifying a link. (Inherited from DiagramBase.)
	LinkPasted	Raised when a link has been pasted from the clipboard. (Inherited from DiagramBase.)
	LinkRouted	Raised when a link is routed by the auto-routing algorithm. (Inherited from DiagramBase.)
	LinkSelected	Raised when a link is selected, either programmatically or by the user. (Inherited from DiagramBase.)
	LinkSelecting	Raised when the user tries to select a link, this event lets you cancel the operation. (Inherited from DiagramBase.)
	LinkSplit	Raised when the user splits a link interactively. (Inherited from DiagramBase.)
	LinkStartModifying	A validation event raised when the user starts modifying a link. (Inherited from DiagramBase.)
	LinkTextEdited	Raised when the user edits the text of a link. (Inherited from DiagramBase.)
	LinkTextEditing	A validation event that lets you approve or reject changes made to the text of a link. (Inherited from DiagramBase.)
	MeasureUnitChanged	Raised when the **MeasureUnit** property value has changed. (Inherited from DiagramBase.)
	NodeActivated	Raised when a node becomes the **ActiveItem**. (Inherited from DiagramBase.)
	NodeClicked	Raised when the user clicks a node. (Inherited from DiagramBase.)
	NodeCreateCancelled	Raised when the user cancels node creation. (Inherited from DiagramBase.)

	NodeCreated	Raised when the user draws a new node. (Inherited from DiagramBase.)
	NodeCreating	A validation event raised while the user is drawing a new node. (Inherited from DiagramBase.)
	NodeDeactivated	Raised when a node is no longer the **ActiveItem**. (Inherited from DiagramBase.)
	NodeDeleted	Raised when a node is deleted, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDeleting	Raised when the user tries to delete a node, this event lets you cancel the operation. (Inherited from DiagramBase.)
	NodeDeselected	Raised when a node is deselected, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeDoubleClicked	Raised when the user double-clicks a node. (Inherited from DiagramBase.)
	NodeHovered	Raised when the mouse pointer hovers over a node. (Inherited from DiagramBase.)
	NodeModified	Raised when the user moves or resizes a node. (Inherited from DiagramBase.)
	NodeModifyCancelled	Raised when the user cancels move or resize operation being done on a node. (Inherited from DiagramBase.)
	NodeModifying	A validation event raised while the user is moving or resizing a node. (Inherited from DiagramBase.)
	NodePasted	Raised when a node has been pasted from the clipboard. (Inherited from DiagramBase.)
	NodeSelected	Raised when a node is selected, either programmatically or by the user. (Inherited from DiagramBase.)
	NodeSelecting	Raised when the user tries to select a node, this event lets you cancel the operation. (Inherited from DiagramBase.)
	NodeStartModifying	A validation event raised when the user starts modifying a node. (Inherited from DiagramBase.)
	NodeTextEdited	Raised when the user edits the text of a node. (Inherited from DiagramBase.)
	NodeTextEditing	A validation event that lets you approve or reject changes made to the text of a node. (Inherited from DiagramBase.)
	OriginAnchorChanged	Raised when the OriginAnchor property value of a link has changed. (Inherited from DiagramBase.)
	PropertyChanged	**INotifyPropertyChanged.PropertyChanged** implementation.
	QueryAlignTarget	Raised to validate a node as target for alignment guides when **AutoAlignNodes** is enabled. (Inherited from DiagramBase.)
	Repaint	Raised when the diagram views should be repainted. (Inherited from DiagramBase.)
	RequestCacheReset	Raised when an object within the diagram requests a reset of the cached bitmap. (Inherited from DiagramBase.)
	RequestClipboard	Raised when an object within the diagram requests an operation with the clipboard. (Inherited from DiagramBase.)
	RequestDrawInteractions	For internal use. (Inherited from DiagramBase.)
	RequestTimer	Raised when an object within the diagram needs to start or stop a timer. (Inherited from DiagramBase.)
	SelectionChanged	Raised when an item is added to or removed from the selection (Inherited from DiagramBase.)

	SelectionModified	Raised when the user moves multiple selected items. (Inherited from DiagramBase.)
	SelectionModifying	Raised while the user is moving multiple selected items, this event lets you cancel the modification. (Inherited from DiagramBase.)
	SerializeTag	Raised when the **Tag** of a diagram item must be serialized into an XML document. (Inherited from DiagramBase.)
	SetSelfLoopShape	Raised when a link becomes a self-loop, giving you a chance to override the default look shape. (Inherited from DiagramBase.)
	ShapeModified	Raised when the user modifies the shape of a ShapeNode by moving one of the control points corresponding to shape formula parameters. (Inherited from DiagramBase.)
	ShapeNotDefined	Raised when the diagram being loaded contains nodes with an undefined shape. (Inherited from DiagramBase.)
	TableColumnResized	Raised when the user finishes resizing a column in a table. (Inherited from DiagramBase.)
	TableColumnResizing	Raised when the user resizes a column in a table interactively. (Inherited from DiagramBase.)
	TableRowResized	Raised when the user finishes resizing a row in a table. (Inherited from DiagramBase.)
	TableRowResizing	Raised when the user resizes a row in a table interactively. (Inherited from DiagramBase.)
	TableSectionCollapsed	Raised when the user collapses the rows of a table's section by clicking the [-] button in the section's header row. (Inherited from DiagramBase.)
	TableSectionExpanded	Raised when the user expands the rows of a table's section by clicking the [+] button in the section's header row. (Inherited from DiagramBase.)
	TreeCollapsed	Raised when a user collapses a tree branch by clicking the [-] button of an **Expandable** node. (Inherited from DiagramBase.)
	TreeExpanded	Raised when a user expands a tree branch by clicking the [+] button of an **Expandable** node. (Inherited from DiagramBase.)
	TreeItemTextEditing	A validation event that lets you approve or reject changes made to the text of a TreeViewNode's item. (Inherited from DiagramBase.)
	ValidateAnchorPoint	A validation event that lets you approve or reject connecting a link to an anchor point. (Inherited from DiagramBase.)
	ValidateLinkCrossing	A validation event that lets you approve or reject displaying a decoration at the intersection point of links. (Inherited from DiagramBase.)
	ViewRemoved	Raised when a DiagramView is detached from this diagram. (Inherited from DiagramBase.)

See Also

DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage Constructor

Initializes a new instance of the DiagramPage class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramPage ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage Methods

DiagramPage.Clone Method

Creates a copy of this diagram page.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Diagram Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Diagram
```

Return Value

See Also

DiagramPage Members
DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage.CopyTo Method

Copies this page's properties and items to specified target diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void CopyTo (
    Diagram target
)
```

Visual Basic Copy Code

```
Public Overrides Sub CopyTo( _
    target As Diagram _
)
```

Parameters

target
The target diagram.

See Also

DiagramPage Members
DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage.OnPropertyChanged Method

Raises the PropertyChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnPropertyChanged (
    PropertyChangedEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnPropertyChanged( _
    e As PropertyChangedEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

DiagramPage Members
DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage Properties

DiagramPage.Icon Property

Gets or sets an image associated with the page.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage Icon { get; set; }
```

Visual Basic Copy Code

```
Public Property Icon As IImage
```

Property Value

A **Microsoft.Maui.Graphics.IImage** object.

See Also

DiagramPage Members
DiagramPage Class

DiagramPage.IconUrl Property

Gets or sets the URL of the image associated with the page.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string IconUrl { get; set; }
```

Visual Basic Copy Code

```
Public Property IconUrl As String
```

Property Value

See Also

DiagramPage Members
DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage.Title Property

Gets or sets the page title.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Title { get; set; }
```

Visual Basic Copy Code

```
Public Property Title As String
```

Property Value

A string value.

See Also

DiagramPage Members
DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage.TitleColor Property

Gets or sets the color of the title when the title is displayed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Color TitleColor { get; set; }
```

Visual Basic Copy Code

```
Public Property TitleColor As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** instance.

See Also

DiagramPage Members
DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPage Events

DiagramPage.PropertyChanged Event

INotifyPropertyChanged.PropertyChanged implementation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event PropertyChangedEventHandler PropertyChanged
```

Visual Basic Copy Code

```
Public Event PropertyChanged As PropertyChangedEventHandler
```

Event Data

PropertyChanged event handlers receive an argument of type **PropertyChangedEventArgs**.

See Also

DiagramPage Members
DiagramPage Class
MindFusion.Diagramming Namespace

DiagramPageEventArgs Class

Contains the arguments passed to DiagramPage-related event handlers.

Syntax

C# Copy Code

```
public class DiagramPageEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class DiagramPageEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
System.EventArgs
MindFusion.Diagramming.DiagramPageEventArgs
MindFusion.Diagramming.DiagramPageValidationEventArgs

See Also

DiagramPageEventArgs Members
MindFusion.Diagramming Namespace

DiagramPageEventArgs Members

Contains the arguments passed to DiagramPage-related event handlers.

The following tables list the members exposed by the DiagramPageEventArgs type.

Public Constructors

	Name	Description
	DiagramPageEventArgs	Initializes a new instance of the **DiagramPageEventArgs** class.

Public Properties

	Name	Description
	DiagramPage	Gets the **DiagramPage** related to the event.

See Also

DiagramPageEventArgs Class
MindFusion.Diagramming Namespace

DiagramPageEventArgs Constructor

Initializes a new instance of the DiagramPageEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramPageEventArgs (
    DiagramPage diagramPage
)
```

Visual Basic Copy Code

```
Public New ( _
    diagramPage As DiagramPage _
)
```

Parameters

diagramPage
The DiagramPage related to the event.

See Also

DiagramPageEventArgs Class
MindFusion.Diagramming Namespace

DiagramPageEventArgs Properties

DiagramPageEventArgs.DiagramPage Property

Gets the DiagramPage related to the event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramPage DiagramPage { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property DiagramPage As DiagramPage
```

Property Value

A DiagramPage object.

See Also

DiagramPageEventArgs Members
DiagramPageEventArgs Class
MindFusion.Diagramming Namespace

DiagramPageValidationEventArgs Class

Contains the arguments passed to handlers of DiagramPage validation events.

Syntax

C# Copy Code

```
public class DiagramPageValidationEventArgs : DiagramPageEventArgs
```

Visual Basic Copy Code

```
Public Class DiagramPageValidationEventArgs
    Inherits DiagramPageEventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.DiagramPageEventArgs
      MindFusion.Diagramming.DiagramPageValidationEventArgs

See Also

DiagramPageValidationEventArgs Members
MindFusion.Diagramming Namespace

DiagramPageValidationEventArgs Members

Contains the arguments passed to handlers of DiagramPage validation events.

The following tables list the members exposed by the DiagramPageValidationEventArgs type.

Public Constructors

	Name	Description
	DiagramPageValidationEventArgs	Initializes a new instance of the **DiagramPageValidationEventArgs** class.

Public Properties

	Name	Description
	Cancel	Gets or sets a value indicating whether to cancel the current operation.
	DiagramPage	Gets the **DiagramPage** related to the event. (Inherited from DiagramPageEventArgs.)

See Also

DiagramPageValidationEventArgs Class
MindFusion.Diagramming Namespace

DiagramPageValidationEventArgs Constructor

Initializes a new instance of the DiagramPageValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramPageValidationEventArgs (
    DiagramPage diagramPage
)
```

Visual Basic Copy Code

```
Public New ( _
    diagramPage As DiagramPage _
)
```

Parameters

diagramPage
The DiagramPage related to the event.

See Also

DiagramPageValidationEventArgs Class
MindFusion.Diagramming Namespace

DiagramPageValidationEventArgs Properties

DiagramPageValidationEventArgs.Cancel Property

Gets or sets a value indicating whether to cancel the current operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Cancel { get; set; }
```

Visual Basic — Copy Code

```
Public Property Cancel As Boolean
```

Property Value

true to cancel the operation; **false** to confirm it.

Remarks

Set this property to **true** to prevent the user from completing the operation.

See Also

DiagramPageValidationEventArgs Members
DiagramPageValidationEventArgs Class
MindFusion.Diagramming Namespace

DiagramStyle Class

Represents a reusable set of properties that define the appearance of diagrams.

Syntax

C# — Copy Code

```
public class DiagramStyle : Style
```

Visual Basic — Copy Code

```
Public Class DiagramStyle
    Inherits Style
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramStyle

See Also

DiagramStyle Members
MindFusion.Diagramming Namespace

DiagramStyle Members

Represents a reusable set of properties that define the appearance of diagrams.

The following tables list the members exposed by the DiagramStyle type.

Public Constructors

	Name	Description
	DiagramStyle	Initializes a new instance of the **DiagramStyle** class.

Public Properties

	Name	Description
	BackBrush	Gets or sets the brush used for painting the diagram background.
	Brush	Gets or sets the brush used to fill the interior of items.
	FontFamily	Gets or sets the font family of the item texts.

	FontSize	Gets or sets the font size of the item texts.
	FontStyle	Gets or sets the font style of the item texts.
	FontUnit	Gets or sets the font unit of the item texts.
	NodeEffects	Gets a list with all effects applied to the nodes of this diagram.
	ShadowBrush	Gets or sets the brush used to paint the shadow of the items.
	Stroke	Gets or sets the brush used to paint the outline of the items.
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items.
	StrokeThickness	Gets or sets the thickness of the items' outline, expressed in the current units of measure.
	TextBrush	Gets or sets the brush used to paint texts inside the items.

Public Methods

	Name	Description
	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
S	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JSON object.
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)
	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a JsonObject.
	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle Constructor

Initializes a new instance of the DiagramStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramStyle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle Methods

DiagramStyle.LoadFromJson Method

Loads the style content from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the style's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.SaveToJson Method

Saves the style content into a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject the style's data should be stored into.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle Properties

DiagramStyle.BackBrush Property

Gets or sets the brush used for painting the diagram background.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush BackBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property BackBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.Brush Property

Gets or sets the brush used to fill the interior of items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush Brush { get; set; }
```

Visual Basic Copy Code

```
Public Property Brush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.FontFamily Property

Gets or sets the font family of the item texts.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string FontFamily { get; set; }
```

Visual Basic Copy Code

```
Public Property FontFamily As String
```

Property Value

A string value.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.FontSize Property

Gets or sets the font size of the item texts.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double FontSize { get; set; }
```

Visual Basic Copy Code

```
Public Property FontSize As Double
```

Property Value

A **double** value.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.FontStyle Property

Gets or sets the font style of the item texts.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public FontStyle FontStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property FontStyle As FontStyle
```

Property Value

One or more of the members of the **FontStyle** enumeration.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.FontUnit Property

Gets or sets the font unit of the item texts.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public GraphicsUnit FontUnit { get; set; }
```

Visual Basic Copy Code

```
Public Property FontUnit As GraphicsUnit
```

Property Value

A member of the **GraphicsUnit** enumeration.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.NodeEffects Property

Gets a list with all effects applied to the nodes of this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IList<NodeEffect> NodeEffects { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NodeEffects As IList(Of NodeEffect)
```

Property Value

A list of NodeEffect objects.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.ShadowBrush Property

Gets or sets the brush used to paint the shadow of the items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush ShadowBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property ShadowBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.Stroke Property

Gets or sets the brush used to paint the outline of the items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush Stroke { get; set; }
```

Visual Basic

```
Public Property Stroke As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.StrokeDashStyle Property

Gets or sets the pattern of dashes and gaps used to outline items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public DashStyle StrokeDashStyle { get; set; }
```

Visual Basic

```
Public Property StrokeDashStyle As DashStyle
```

Property Value

A member of the **DashStyle** enumeration.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.StrokeThickness Property

Gets or sets the thickness of the items' outline, expressed in the current units of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double StrokeThickness { get; set; }
```

Visual Basic

```
Public Property StrokeThickness As Double
```

Property Value

A double value.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle.TextBrush Property

Gets or sets the brush used to paint texts inside the items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush TextBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property TextBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

DiagramStyle Members
DiagramStyle Class
MindFusion.Diagramming Namespace

DiagramStyle Events

DirectionConstraint Enumeration

Defines values that specify in what direction to constrain node movement.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum DirectionConstraint
```

Visual Basic Copy Code

```
Public Enum DirectionConstraint
```

Members

	Member name	Description
	None	No restrictions.
	Horizontal	Users are allowed to move the node horizontally.
	Vertical	Users are allowed to move the node vertically.

Remarks

Values from this enumeration can be assigned to the MoveDirection property of the NodeConstraints class.

See Also

MindFusion.Diagramming Namespace

DraggedItem Class

The base class of DraggedNode ang DraggedLink classes.

Syntax

C# Copy Code

```
public abstract class DraggedItem
```

Visual Basic Copy Code

```
Public MustInherit Class DraggedItem
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DraggedItem
    MindFusion.Diagramming.DraggedLink
    MindFusion.Diagramming.DraggedNode

See Also

DraggedItem Members
MindFusion.Diagramming Namespace

DraggedItem Members

The base class of DraggedNode ang DraggedLink classes.

The following tables list the members exposed by the DraggedItem type.

Public Constructors

	Name	Description
	DraggedItem	Initializes a new instance of the DraggedItem class.

Public Properties

	Name	Description
	Diagram	The Diagram this item is dropped on.
	Item	The dragged item.
	Label	The text to be displayed for the item.
	Size	The size of the item.

See Also

DraggedItem Class
MindFusion.Diagramming Namespace

DraggedItem Constructor

Initializes a new instance of the DraggedItem class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

Copy Code

```
public DraggedItem (
    DiagramItem item,
    Size size,
    string label = ""
)
```

Visual Basic Copy Code

```
Public New ( _
    item As DiagramItem, _
    size As Size, _
    Optional label As String = "" _
)
```

Parameters

item
The dragged DiagramItem.

size
The size of the item.

label
Optional.
The label to be displayed for the item.

See Also

DraggedItem Class
MindFusion.Diagramming Namespace

DraggedItem Properties

DraggedItem.Diagram Property

The Diagram this item is dropped on.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Diagram Diagram { get; set; }
```

Visual Basic Copy Code

```
Public Property Diagram As Diagram
```

Property Value

See Also

DraggedItem Members
DraggedItem Class
MindFusion.Diagramming Namespace

DraggedItem.Item Property

The dragged item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem Item { get; set; }
```

Visual Basic Copy Code

```
Public Property Item As DiagramItem
```

Property Value

See Also

DraggedItem Members
DraggedItem Class
MindFusion.Diagramming Namespace

DraggedItem.Label Property

The text to be displayed for the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Label { get; set; }
```

Visual Basic Copy Code

```
Public Property Label As String
```

Property Value

See Also

DraggedItem Members
DraggedItem Class
MindFusion.Diagramming Namespace

DraggedItem.Size Property

The size of the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size Size { get; set; }
```

Visual Basic Copy Code

```
Public Property Size As Size
```

Property Value

See Also

DraggedLink Class

Class used to hold the DiagramLink instance during drag and drop. It is used for easier detection of the dragged content as DragEventArgs.Data.GetData() does not check super classes.

Syntax

C# Copy Code

```
public class DraggedLink : DraggedItem
```

Visual Basic Copy Code

```
Public Class DraggedLink
    Inherits DraggedItem
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DraggedItem
    MindFusion.Diagramming.DraggedLink

See Also

DraggedLink Members
MindFusion.Diagramming Namespace

DraggedLink Members

Class used to hold the DiagramLink instance during drag and drop. It is used for easier detection of the dragged content as DragEventArgs.Data.GetData() does not check super classes.

The following tables list the members exposed by the DraggedLink type.

Public Constructors

	Name	Description
	DraggedLink	Initializes a new instance of the DraggedLink class.

Public Properties

	Name	Description
	AutoConnectDistance	
	AutoConnectLinks	
	BaseShape	
	BaseShapeSize	
	Diagram	The Diagram this item is dropped on. (Inherited from DraggedItem.)
	HeadPen	
	HeadShape	
	HeadShapeSize	
	Item	The dragged item. (Inherited from DraggedItem.)

	Label	The text to be displayed for the item. (Inherited from DraggedItem.)
	Link	Returns the dragged DiagramLink.
	Size	The size of the item. (Inherited from DraggedItem.)

Public Methods

	Name	Description
	Connect	Overloaded.
	RestoreState	
	UpdateState	

See Also

DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink Constructor

Initializes a new instance of the DraggedLink class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DraggedLink (
    DiagramLink link,
    Size size,
    string label
)
```

Visual Basic Copy Code

```
Public New ( _
    link As DiagramLink, _
    size As Size, _
    label As String _
)
```

Parameters

link
The dragged DiagramLink.

size
The size of the link.

label
The text to be displayed for the link.

See Also

DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink Methods

DraggedLink.Connect Method

Overload List

Name	Description
DraggedLink.Connect (DiagramLink, Point)	
DraggedLink.Connect (DiagramLink, Point, Boolean, Boolean@, Boolean@)	

See Also

DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink.Connect Method (DiagramLink, Point)

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Connect (
    DiagramLink link,
    Point point
)
```

Visual Basic Copy Code

```
Public Sub Connect( _
    link As DiagramLink, _
    point As Point _
)
```

Parameters

link

point

See Also

Connect Method Overload List
DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink.Connect Method (DiagramLink, Point, Boolean, Boolean@, Boolean@)

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Connect (
    DiagramLink link,
```

```
    Point point,
    bool doConnect,
    ref bool origin,
    ref bool destination
)
```

Visual Basic Copy Code

```
Public Sub Connect( _
    link As DiagramLink, _
    point As Point, _
    doConnect As Boolean, _
    ByRef origin As Boolean, _
    ByRef destination As Boolean _
)
```

Parameters

link

point

doConnect

origin

destination

See Also

Connect Method Overload List
DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink.RestoreState Method

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RestoreState (
    bool destination = false
)
```

Visual Basic Copy Code

```
Public Sub RestoreState( _
    Optional destination As Boolean = False _
)
```

Parameters

destination
Optional.

See Also

DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink.UpdateState Method

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void UpdateState (
    bool destination = false
)
```

Visual Basic Copy Code

```
Public Sub UpdateState( _
    Optional destination As Boolean = False _
)
```

Parameters

destination
Optional.

See Also

DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink Properties

DraggedLink.AutoConnectDistance Property

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int AutoConnectDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoConnectDistance As Integer
```

Property Value

See Also

DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink.AutoConnectLinks Property

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoConnectLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoConnectLinks As Boolean
```

Property Value

See Also

DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink.BaseShape Property

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape BaseShape { get; set; }
```

Visual Basic Copy Code

```
Public Property BaseShape As Shape
```

Property Value

See Also

DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink.BaseShapeSize Property

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double BaseShapeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property BaseShapeSize As Double
```

Property Value

See Also

DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink.HeadPen Property

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen HeadPen { get; set; }
```

Visual Basic Copy Code

```
Public Property HeadPen As Pen
```

Property Value

See Also

DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink.HeadShape Property

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape HeadShape { get; set; }
```

Visual Basic Copy Code

```
Public Property HeadShape As Shape
```

Property Value

See Also

DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink.HeadShapeSize Property

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double HeadShapeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property HeadShapeSize As Double
```

Property Value

See Also

DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedLink.Link Property

Returns the dragged DiagramLink.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink Link { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Link As DiagramLink
```

Property Value

See Also

DraggedLink Members
DraggedLink Class
MindFusion.Diagramming Namespace

DraggedNode Class

Class used to hold the DiagramNode instance during drag and drop. It is used for easier detection of the dragged content as **DragEventArgs.Data.GetData** does not check super classes.

Syntax

C# Copy Code

```
public class DraggedNode : DraggedItem
```

Visual Basic Copy Code

```
Public Class DraggedNode
    Inherits DraggedItem
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DraggedItem
    MindFusion.Diagramming.DraggedNode

See Also

DraggedNode Members
MindFusion.Diagramming Namespace

DraggedNode Members

Class used to hold the DiagramNode instance during drag and drop. It is used for easier detection of the dragged content as **DragEventArgs.Data.GetData** does not check super classes.

The following tables list the members exposed by the DraggedNode type.

Public Constructors

	Name	Description
	DraggedNode	Overloaded. Initializes a new instance of the **DraggedNode** class.

Public Properties

	Name	Description
	Diagram	The Diagram this item is dropped on. (Inherited from DraggedItem.)
	Item	The dragged item. (Inherited from DraggedItem.)
	Label	The text to be displayed for the node.
	Node	Holds the dragged DiagramNode.
	Size	The size of the node.

See Also

DraggedNode Class
MindFusion.Diagramming Namespace

DraggedNode Constructor

Initializes a new instance of the DraggedNode class.

Overload List

Name	Description
DraggedNode (DiagramNode, Size)	Initializes a new instance of the **DraggedNode** class.
DraggedNode (DiagramNode, Size, String)	Initializes a new instance of the **DraggedNode** class.

See Also

DraggedNode Class
MindFusion.Diagramming Namespace

DraggedNode Constructor (DiagramNode, Size)

Initializes a new instance of the DraggedNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DraggedNode (
    DiagramNode node,
    Size size
)
```

Visual Basic

```
Public New ( _
    node As DiagramNode, _
    size As Size _
)
```

Parameters

node
The dragged DiagramNode.

size
The size of the node.

See Also

DraggedNode Constructor Overload List
DraggedNode Class
MindFusion.Diagramming Namespace

DraggedNode Constructor (DiagramNode, Size, String)

Initializes a new instance of the DraggedNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public DraggedNode (
    DiagramNode node,
    Size size,
    string label
)
```

Visual Basic

```
Public New ( _
    node As DiagramNode, _
    size As Size, _
    label As String _
)
```

Parameters

node
The dragged DiagramNode.

size
The size of the node.

label
The text to be displayed for the node.

See Also

DraggedNode Constructor Overload List
DraggedNode Class
MindFusion.Diagramming Namespace

DraggedNode Properties

DraggedNode.Label Property

The text to be displayed for the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Label { get; set; }
```

Visual Basic Copy Code

```
Public Property Label As String
```

Property Value

A string value.

See Also

DraggedNode Members
DraggedNode Class
MindFusion.Diagramming Namespace

DraggedNode.Node Property

Holds the dragged DiagramNode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Node { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Node As DiagramNode
```

Property Value

A DiagramNode object.

See Also

DraggedNode Members
DraggedNode Class
MindFusion.Diagramming Namespace

DraggedNode.Size Property

The size of the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size Size { get; set; }
```

Visual Basic Copy Code

```
Public Property Size As Size
```

Property Value

A **Microsoft.Maui.Graphics.Size** instance.

See Also

DraggedNode Members
DraggedNode Class
MindFusion.Diagramming Namespace

DraggedShape Class

Contains drag-and-drop information for a shape being dragged from **ShapeToolBar** and **ShapeListBox**.

Syntax

C# Copy Code

```
public class DraggedShape
```

Visual Basic Copy Code

```
Public Class DraggedShape
```

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.DraggedShape

See Also

DraggedShape Members
MindFusion.Diagramming Namespace

DraggedShape Members

Contains drag-and-drop information for a shape being dragged from **ShapeToolBar** and **ShapeListBox**.

The following tables list the members exposed by the DraggedShape type.

Public Constructors

	Name	Description
	DraggedShape	Initializes a new instance of the **DraggedShape** class.

Public Properties

	Name	Description
	Id	Gets the Id of the dragged shape.
	Size	Gets the recommended size of the dragged shape.

See Also

DraggedShape Class
MindFusion.Diagramming Namespace

DraggedShape Constructor

Initializes a new instance of the DraggedShape class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DraggedShape (
    string id,
    Size size
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String, _
    size As Size _
)
```

Parameters

id
The Id of the dragged shape.

size
The recommended size of the dragged shape.

See Also

DraggedShape Class
MindFusion.Diagramming Namespace

DraggedShape Properties

DraggedShape.Id Property

Gets the Id of the dragged shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Id { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Id As String
```

Property Value

A string specifying the identifier of the dragged shape.

See Also

DraggedShape Members
DraggedShape Class
MindFusion.Diagramming Namespace

DraggedShape.Size Property

Gets the recommended size of the dragged shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size Size { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Size As Size
```

Property Value

A **Microsoft.Maui.Graphics.Size** specifying the recommended size for the shape.

See Also

DraggedShape Members
DraggedShape Class
MindFusion.Diagramming Namespace

DragIndicator Enumeration

Specifies what kind of feedback to display during drag-and-drop from NodeListView.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum DragIndicator
```

Visual Basic Copy Code

```
Public Enum DragIndicator
```

Members

	Member name	Description
	SystemDefault	Show default system cursor.
	GhostImage	Draw the actual dragged node on the DiagramView surface.
	MouseCursor	Change the mouse cursor to an icon representing the dragged node.

See Also

MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs Class

Contains the arguments passed to the DrawAnchorPoint event handlers.

Syntax

C# Copy Code

```
public class DrawAnchorPointEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class DrawAnchorPointEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.DrawAnchorPointEventArgs

See Also

DrawAnchorPointEventArgs Members
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs Members

Contains the arguments passed to the DrawAnchorPoint event handlers.

The following tables list the members exposed by the DrawAnchorPointEventArgs type.

Public Constructors

	Name	Description
	DrawAnchorPointEventArgs	Initializes a new instance of the **DrawAnchorPointEventArgs** class.

Public Properties

	Name	Description
	AnchorIndex	Gets the index of the anchor point within its containing pattern.
	AnchorPattern	Gets the anchor pattern containing the anchor point being drawn.
	Graphics	Gets an object implementing the IGraphics interface whose methods should be used to draw the anchor point mark.
	Location	Gets the location of the anchor point being drawn.
	Node	Gets the node whose anchor point is being drawn.

See Also

DrawAnchorPointEventArgs Class
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs Constructor

Initializes a new instance of the DrawAnchorPointEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DrawAnchorPointEventArgs (
    IGraphics graphics,
    Point location,
```

```
    DiagramNode node,
    AnchorPattern anchorPattern,
    int anchorIndex
)
```

Visual Basic Copy Code

```
Public New ( _
    graphics As IGraphics, _
    location As Point, _
    node As DiagramNode, _
    anchorPattern As AnchorPattern, _
    anchorIndex As Integer _
)
```

Parameters

graphics
The IGraphics object that represents the target of the drawing operation.

location
The position of the anchor point, expressed in document coordinates.

node
The DiagramNode related to the event.

anchorPattern
The AnchorPattern object whose point is being drawn.

anchorIndex
The zero-based index of the anchor point being drawn within its anchor pattern.

See Also

DrawAnchorPointEventArgs Class
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs Properties

DrawAnchorPointEventArgs.AnchorIndex Property

Gets the index of the anchor point within its containing pattern.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int AnchorIndex { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property AnchorIndex As Integer
```

Property Value

An integer value specifying the zero-based index of the anchor point within its AnchorPattern.

See Also

DrawAnchorPointEventArgs Members
DrawAnchorPointEventArgs Class
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs.AnchorPattern Property

Gets the anchor pattern containing the anchor point being drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern AnchorPattern { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property AnchorPattern As AnchorPattern
```

Property Value

An instance of the AnchorPattern class.

See Also

DrawAnchorPointEventArgs Members
DrawAnchorPointEventArgs Class
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs.Graphics Property

Gets an object implementing the IGraphics interface whose methods should be used to draw the anchor point mark.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IGraphics Graphics { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Graphics As IGraphics
```

Property Value

An instance of a class implementing the IGraphics interface.

See Also

DrawAnchorPointEventArgs Members
DrawAnchorPointEventArgs Class
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs.Location Property

Gets the location of the anchor point being drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point Location { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Location As Point
```

Property Value

A **Microsoft.Maui.Graphics.Point** specifying the location of the anchor point.

See Also

DrawAnchorPointEventArgs Members
DrawAnchorPointEventArgs Class
MindFusion.Diagramming Namespace

DrawAnchorPointEventArgs.Node Property

Gets the node whose anchor point is being drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Node { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Node As DiagramNode
```

Property Value

An instance of the DiagramNode class.

See Also

DrawAnchorPointEventArgs Members
DrawAnchorPointEventArgs Class
MindFusion.Diagramming Namespace

DrawBranchIndicatorEventArgs Class

Contains the arguments passed to the DrawBranchIndicator event handlers.

Syntax

C# Copy Code

```
public class DrawBranchIndicatorEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class DrawBranchIndicatorEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.DrawBranchIndicatorEventArgs

See Also

DrawBranchIndicatorEventArgs Members
MindFusion.Diagramming Namespace

DrawBranchIndicatorEventArgs Members

Contains the arguments passed to the DrawBranchIndicator event handlers.

The following tables list the members exposed by the DrawBranchIndicatorEventArgs type.

Public Properties

	Name	Description
	From	A Vector specifying the direction of the incoming link segment.
	Graphics	Gets an IGraphics object whose methods should be used to draw the branch indicator.
	Merge	A value indicating whether the indicator designates merge or branch point.
	Position	A **Point** specifying the position of the branch indicator.
	To	A **Vector** specifying the direction of the outgoing link segment.

See Also

DrawBranchIndicatorEventArgs Class
MindFusion.Diagramming Namespace

DrawBranchIndicatorEventArgs Properties

DrawBranchIndicatorEventArgs.From Property

A Vector specifying the direction of the incoming link segment.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Vector From { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property From As Vector
```

Property Value

A Vector instance specifying the link direction before the branch point's bend.

See Also

DrawBranchIndicatorEventArgs Members
DrawBranchIndicatorEventArgs Class
MindFusion.Diagramming Namespace

DrawBranchIndicatorEventArgs.Graphics Property

Gets an IGraphics object whose methods should be used to draw the branch indicator.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IGraphics Graphics { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Graphics As IGraphics
```

Property Value

An object implementing the IGraphics interface.

See Also

DrawBranchIndicatorEventArgs Members
DrawBranchIndicatorEventArgs Class
MindFusion.Diagramming Namespace

DrawBranchIndicatorEventArgs.Merge Property

A value indicating whether the indicator designates merge or branch point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Merge { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Merge As Boolean
```

Property Value

true to draw a merge indicator, or **false** to draw a branch indicator.

See Also

DrawBranchIndicatorEventArgs Members
DrawBranchIndicatorEventArgs Class
MindFusion.Diagramming Namespace

DrawBranchIndicatorEventArgs.Position Property

A **Point** specifying the position of the branch indicator.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point Position { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Position As Point
```

Property Value

A **Microsoft.Maui.Graphics.Point** containing coordinates of the indicator.

See Also

DrawBranchIndicatorEventArgs Members
DrawBranchIndicatorEventArgs Class
MindFusion.Diagramming Namespace

DrawBranchIndicatorEventArgs.To Property

A Vector specifying the direction of the outgoing link segment.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Vector To { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property To As Vector
```

Property Value

A Vector instance specifying the link direction after the branch point's bend.

See Also

DrawBranchIndicatorEventArgs Members
DrawBranchIndicatorEventArgs Class
MindFusion.Diagramming Namespace

DrawCellEventArgs Class

Contains the arguments passed to the DrawCell event handlers.

Syntax

C# Copy Code

```
public class DrawCellEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class DrawCellEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.DrawCellEventArgs

See Also

DrawCellEventArgs Members
MindFusion.Diagramming Namespace

DrawCellEventArgs Members

Contains the arguments passed to the DrawCell event handlers.

The following tables list the members exposed by the DrawCellEventArgs type.

Public Constructors

	Name	Description
	DrawCellEventArgs	Initializes a new instance of the **DrawCellEventArgs** class.

Public Properties

	Name	Description
	Bounds	Gets the bounding rectangle of the cell being drawn.
	Cell	Gets the cell that should be custom drawn.
	Column	Gets the index of the table's column that contains **Cell**.
	Graphics	An object implementing the IGraphics interface whose methods should be used to draw the cell.
	Row	Gets the index of the table's row that contains **Cell**.
	Table	Get the table that contains **Cell**.

See Also

DrawCellEventArgs Class
MindFusion.Diagramming Namespace

DrawCellEventArgs Constructor

Initializes a new instance of the DrawCellEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DrawCellEventArgs (
    IGraphics graphics,
    TableNode table,
    int column,
    int row,
    Rect bounds
)
```

Visual Basic Copy Code

```
Public New ( _
    graphics As IGraphics, _
    table As TableNode, _
    column As Integer, _
    row As Integer, _
    bounds As Rect _
)
```

Parameters

graphics
The IGraphics object that represents the target of the drawing operation.

table
The TableNode related to the event.

column
The zero-based index of the table column related to the event.

row
The zero-based index of the table row related to the event.

bounds
The bounding rectangle of the cell being drawn.

See Also

DrawCellEventArgs Class
MindFusion.Diagramming Namespace

DrawCellEventArgs Properties

DrawCellEventArgs.Bounds Property

Gets the bounding rectangle of the cell being drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect Bounds { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Bounds As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** specifying the bounding rectangle of the cell.

See Also

DrawCellEventArgs Members
DrawCellEventArgs Class
MindFusion.Diagramming Namespace

DrawCellEventArgs.Cell Property

Gets the cell that should be custom drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Cell Cell { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Cell As TableNode.Cell
```

Property Value

An instance of the TableNode.Cell class.

See Also

DrawCellEventArgs Members
DrawCellEventArgs Class
MindFusion.Diagramming Namespace

DrawCellEventArgs.Column Property

Gets the index of the table's column that contains Cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Column { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Column As Integer
```

Property Value

An integer value specifying the zero-based index of the column.

See Also

DrawCellEventArgs Members
DrawCellEventArgs Class
MindFusion.Diagramming Namespace

DrawCellEventArgs.Graphics Property

An object implementing the IGraphics interface whose methods should be used to draw the cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IGraphics Graphics { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Graphics As IGraphics
```

Property Value

An instance of a class implementing the IGraphics interface.

See Also

DrawCellEventArgs Members
DrawCellEventArgs Class
MindFusion.Diagramming Namespace

DrawCellEventArgs.Row Property

Gets the index of the table's row that contains Cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Row { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Row As Integer
```

Property Value

An integer value specifying the zero-based index of the row.

See Also

DrawCellEventArgs Members
DrawCellEventArgs Class
MindFusion.Diagramming Namespace

DrawCellEventArgs.Table Property

Get the table that contains Cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode Table { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Table As TableNode
```

Property Value

An instance of the TableNode class.

See Also

DrawCellEventArgs Members
DrawCellEventArgs Class
MindFusion.Diagramming Namespace

DrawItemEventArgs Class

Contains the arguments passed to the DrawAdjustmentHandles event handlers.

Syntax

C# Copy Code

```
public class DrawItemEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class DrawItemEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.DrawItemEventArgs

See Also

DrawItemEventArgs Members
MindFusion.Diagramming Namespace

DrawItemEventArgs Members

Contains the arguments passed to the DrawAdjustmentHandles event handlers.

The following tables list the members exposed by the DrawItemEventArgs type.

Public Constructors

	Name	Description
	DrawItemEventArgs	Initializes a new instance of the **DrawItemEventArgs** class.

Public Properties

	Name	Description
	Graphics	Gets an object implementing the IGraphics interface whose methods should be used to draw the adjustment handles.
	Item	Gets the item whose selection handles are being drawn.

See Also

DrawItemEventArgs Class
MindFusion.Diagramming Namespace

DrawItemEventArgs Constructor

Initializes a new instance of the DrawItemEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DrawItemEventArgs (
    IGraphics graphics,
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public New ( _
    graphics As IGraphics, _
    item As DiagramItem _
)
```

Parameters

graphics
The IGraphics object that represents the target of the drawing operation.

item
The DiagramItem related to the event.

See Also

DrawItemEventArgs Class
MindFusion.Diagramming Namespace

DrawItemEventArgs Properties

DrawItemEventArgs.Graphics Property

Gets an object implementing the IGraphics interface whose methods should be used to draw the adjustment handles.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IGraphics Graphics { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Graphics As IGraphics
```

Property Value

An instance of a class implementing the IGraphics interface.

See Also

DrawItemEventArgs Members
DrawItemEventArgs Class
MindFusion.Diagramming Namespace

DrawItemEventArgs.Item Property

Gets the item whose selection handles are being drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem Item { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Item As DiagramItem
```

Property Value

An instance of the DiagramItem class.

See Also

DrawItemEventArgs Members
DrawItemEventArgs Class
MindFusion.Diagramming Namespace

DrawLinkEventArgs Class

Contains the arguments passed to the DrawLink event handlers.

Syntax

C# Copy Code

```
public class DrawLinkEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class DrawLinkEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.DrawLinkEventArgs

See Also

DrawLinkEventArgs Members
MindFusion.Diagramming Namespace

DrawLinkEventArgs Members

Contains the arguments passed to the DrawLink event handlers.

The following tables list the members exposed by the DrawLinkEventArgs type.

Public Constructors

	Name	Description
	DrawLinkEventArgs	Initializes a new instance of the **DrawLinkEventArgs** class.

Public Properties

	Name	Description
	Graphics	Gets an object implementing the IGraphics interface whose methods should be used to draw the link.
	Link	Gets the DiagramLink object that should be custom drawn.
	Points	Gets a collection of the link's control points.
	Shadow	Gets a value indicating whether the event is raised to draw the link's shadow.

See Also

DrawLinkEventArgs Class
MindFusion.Diagramming Namespace

DrawLinkEventArgs Constructor

Initializes a new instance of the DrawLinkEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DrawLinkEventArgs (
    IGraphics graphics,
    DiagramLink link,
    bool shadow,
    PointCollection points
)
```

Visual Basic Copy Code

```
Public New ( _
    graphics As IGraphics, _
    link As DiagramLink, _
    shadow As Boolean, _
    points As PointCollection _
)
```

Parameters

graphics
The IGraphics object that represents the target of the drawing operation.

link
The DiagramLink related to the event.

shadow
Specifies whether to draw the link's shadow or the link itself.

points
The collection containing the coordinates of the link control points, pretranslated depending on whether drawing the shadow or not.

See Also

DrawLinkEventArgs Class
MindFusion.Diagramming Namespace

DrawLinkEventArgs Properties

DrawLinkEventArgs.Graphics Property

Gets an object implementing the IGraphics interface whose methods should be used to draw the link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IGraphics Graphics { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Graphics As IGraphics
```

Property Value

An instance of a class implementing the IGraphics interface.

See Also

DrawLinkEventArgs Members
DrawLinkEventArgs Class
MindFusion.Diagramming Namespace

DrawLinkEventArgs.Link Property

Gets the DiagramLink object that should be custom drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink Link { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Link As DiagramLink
```

Property Value

An instance of the DiagramLink class.

See Also

DrawLinkEventArgs Members
DrawLinkEventArgs Class
MindFusion.Diagramming Namespace

DrawLinkEventArgs.Points Property

Gets a collection of the link's control points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PointCollection Points { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Points As PointCollection
```

Property Value

An instance of the PointCollection class.

Remarks

When drawing shadows, the points are offset as necessary.

See Also

DrawLinkEventArgs Members
DrawLinkEventArgs Class
MindFusion.Diagramming Namespace

DrawLinkEventArgs.Shadow Property

Gets a value indicating whether the event is raised to draw the link's shadow.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Shadow { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Shadow As Boolean
```

Property Value

true if now drawing the link's shadow; otherwise, **false**.

See Also

DrawLinkEventArgs Members
DrawLinkEventArgs Class
MindFusion.Diagramming Namespace

DrawNodeEventArgs Class

Contains the arguments passed to the DrawNode event handlers.

Syntax

C# Copy Code

```
public class DrawNodeEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class DrawNodeEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.DrawNodeEventArgs

See Also

DrawNodeEventArgs Members
MindFusion.Diagramming Namespace

DrawNodeEventArgs Members

Contains the arguments passed to the DrawNode event handlers.

The following tables list the members exposed by the DrawNodeEventArgs type.

Public Constructors

	Name	Description
	DrawNodeEventArgs	Initializes a new instance of the **DrawNodeEventArgs** class.

Public Properties

	Name	Description
	Bounds	Gets the bounding rectangle of the node being drawn.
	Graphics	Gets an object implementing the IGraphics interface whose methods should be used to draw the node.
	Node	Gets the DiagramNode object that should be custom drawn.
	Shadow	Gets a value indicating whether the event is raised to draw the node's shadow.

See Also

DrawNodeEventArgs Class
MindFusion.Diagramming Namespace

DrawNodeEventArgs Constructor

Initializes a new instance of the DrawNodeEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DrawNodeEventArgs (
    IGraphics graphics,
    DiagramNode node,
    bool shadow,
    Rect bounds
)
```

Visual Basic Copy Code

```
Public New ( _
    graphics As IGraphics, _
    node As DiagramNode, _
    shadow As Boolean, _
    bounds As Rect _
)
```

Parameters

graphics
The IGraphics object that represents the target of the drawing operation.

node
The DiagramNode related to the event.

shadow
Specifies whether to draw the node's shadow or the node itself.

bounds
The bounding rectangle of the node being drawn.

See Also

DrawNodeEventArgs Class
MindFusion.Diagramming Namespace

DrawNodeEventArgs Properties

DrawNodeEventArgs.Bounds Property

Gets the bounding rectangle of the node being drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect Bounds { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Bounds As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** specifying the bounding rectangle of the node being drawn.

Remarks

When drawing shadows, the rectangle is offset as necessary.

See Also

DrawNodeEventArgs Members
DrawNodeEventArgs Class
MindFusion.Diagramming Namespace

DrawNodeEventArgs.Graphics Property

Gets an object implementing the IGraphics interface whose methods should be used to draw the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IGraphics Graphics { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Graphics As IGraphics
```

Property Value

An instance of a class implementing the IGraphics interface.

See Also

DrawNodeEventArgs Members
DrawNodeEventArgs Class
MindFusion.Diagramming Namespace

DrawNodeEventArgs.Node Property

Gets the DiagramNode object that should be custom drawn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Node { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Node As DiagramNode
```

Property Value

An instance of the DiagramNode class.

See Also

DrawNodeEventArgs Members
DrawNodeEventArgs Class
MindFusion.Diagramming Namespace

DrawNodeEventArgs.Shadow Property

Gets a value indicating whether the event is raised to draw the node's shadow.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Shadow { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Shadow As Boolean
```

Property Value

true if now drawing the node's shadow; otherwise, **false**.

See Also

DrawNodeEventArgs Members
DrawNodeEventArgs Class
MindFusion.Diagramming Namespace

DummyNode Class

Represents virtual nodes used for disconnected links.

Syntax

C# Copy Code

```
public class DummyNode : DiagramNode
```

Visual Basic Copy Code

```
Public Class DummyNode
    Inherits DiagramNode
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.DummyNode

See Also

DummyNode Members
MindFusion.Diagramming Namespace

DummyNode Members

Represents virtual nodes used for disconnected links.

The following tables list the members exposed by the DummyNode type.

Public Constructors

	Name	Description
	DummyNode	Initializes a new instance of the **DummyNode** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)

	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)

	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	DiagramItem.ContainsPoint override.
	CreateConnectionPoint	DiagramNode.CreateConnectionPoint override.
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override.
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override.
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override.

	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	DiagramNode.GetIntersection override.
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the smallest distance from the specified point.
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override.
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)

	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)

DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
LoadFromJson	Loads the node from a JSON object. (Inherited from DiagramNode.)
LoadFromXml	DiagramItem.LoadFromXml override. (Inherited from DiagramNode.)
LoadObsolete	For internal use. (Inherited from DiagramItem.)
OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
Repaint	Overloaded. (Inherited from DiagramItem.)
RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)

	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object. (Inherited from DiagramNode.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramNode.)
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

DummyNode Class
MindFusion.Diagramming Namespace

DummyNode Constructor

Initializes a new instance of the DummyNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DummyNode (
    Diagram parent
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram _
)
```

Parameters

parent
The Diagram this node will be associated with.

See Also

DummyNode Class
MindFusion.Diagramming Namespace

DummyNode Methods

DummyNode.ContainsPoint Method

DiagramItem.ContainsPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool ContainsPoint (
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function ContainsPoint( _
    point As Point _
) As Boolean
```

Parameters

point
The point to check.

Return Value

See Also

DummyNode Members
DummyNode Class
MindFusion.Diagramming Namespace

DummyNode.CreateConnectionPoint Method

DiagramNode.CreateConnectionPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override ConnectionPoint CreateConnectionPoint (
    DiagramLink link,
    Point point,
    bool incoming
)
```

Visual Basic Copy Code

```
Public Overrides Function CreateConnectionPoint( _
    link As DiagramLink, _
    point As Point, _
    incoming As Boolean _
) As ConnectionPoint
```

Parameters

link

point

incoming

Return Value

See Also

DummyNode Members
DummyNode Class
MindFusion.Diagramming Namespace

DummyNode.Draw Method

DiagramItem.Draw override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Draw (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub Draw( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
The IGraphics surface to draw the item on.

options
A RenderOptions object specifying options for the drawing operation.

See Also

DummyNode Members
DummyNode Class
MindFusion.Diagramming Namespace

DummyNode.DrawHandles Method

DiagramItem.DrawHandles override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawHandles (
    IGraphics graphics,
    HandlesVisualStyle style,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawHandles( _
    graphics As IGraphics, _
    style As HandlesVisualStyle, _
    options As RenderOptions _
)
```

Parameters

graphics

style

options

See Also

DummyNode Members
DummyNode Class
MindFusion.Diagramming Namespace

DummyNode.DrawShadow Method

DiagramItem.DrawShadow override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawShadow (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawShadow( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
The IGraphics surface to draw the shadow on.

options
A RenderOptions object specifying options for the drawing operation.

See Also

DummyNode Members
DummyNode Class
MindFusion.Diagramming Namespace

DummyNode.GetIntersection Method

DiagramNode.GetIntersection override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Point GetIntersection (
    Point segmentStart,
    Point segmentEnd
)
```

Visual Basic Copy Code

```
Public Overrides Function GetIntersection( _
    segmentStart As Point, _
    segmentEnd As Point _
) As Point
```

Parameters

segmentStart

segmentEnd

Return Value

See Also

DummyNode Members
DummyNode Class
MindFusion.Diagramming Namespace

DummyNode.GetNearestBorderPoint Method

Returns a point that lies on the outline of this node, at the smallest distance from the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Point GetNearestBorderPoint (
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function GetNearestBorderPoint( _
    point As Point _
) As Point
```

Parameters

point

Return Value

See Also

DummyNode Members
DummyNode Class
MindFusion.Diagramming Namespace

DummyNode.HitTestHandle Method

DiagramItem.HitTestHandle override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override AdjustmentHandle HitTestHandle (
    Point point,
    double distance
)
```

Visual Basic Copy Code

```
Public Overrides Function HitTestHandle( _
    point As Point, _
    distance As Double _
) As AdjustmentHandle
```

Parameters

point

distance

Return Value

See Also

DummyNode Members
DummyNode Class
MindFusion.Diagramming Namespace

EditCellTextEventArgs Class

Contains the arguments passed to the CellTextEdited event handlers.

Syntax

C# Copy Code

```
public class EditCellTextEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class EditCellTextEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.EditCellTextEventArgs

See Also

EditCellTextEventArgs Members
MindFusion.Diagramming Namespace

EditCellTextEventArgs Members

Contains the arguments passed to the CellTextEdited event handlers.

The following tables list the members exposed by the EditCellTextEventArgs type.

Public Constructors

	Name	Description
	EditCellTextEventArgs	Initializes a new instance of the **EditCellTextEventArgs** class.

Public Properties

	Name	Description
	Cell	Get the cell whose text has been edited.
	Column	Gets the index of the table's column that contains **Cell**.
	NewText	Gets the text that has just been entered by the user.
	OldText	Gets the text that had been displayed in the cell before the user edited it.
	Row	Gets the index of the table's row that contains **Cell**.
	Table	Get the table that contains **Cell**.

See Also

EditCellTextEventArgs Class
MindFusion.Diagramming Namespace

EditCellTextEventArgs Constructor

Initializes a new instance of the EditCellTextEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public EditCellTextEventArgs (
    TableNode.Cell cell,
    string oldText,
    string newText,
    int column,
    int row
)
```

Visual Basic Copy Code

```
Public New ( _
    cell As TableNode.Cell, _
    oldText As String, _
    newText As String, _
    column As Integer, _
    row As Integer _
)
```

Parameters

cell
The TableNode.Cell related to the event.

oldText
The old text of the cell.

newText
The new text of the cell.

column
The zero-based index of the table column related to the event.

row
The zero-based index of the table row related to the event.

See Also

EditCellTextEventArgs Class
MindFusion.Diagramming Namespace

EditCellTextEventArgs Properties

EditCellTextEventArgs.Cell Property

Get the cell whose text has been edited.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Cell Cell { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Cell As TableNode.Cell
```

Property Value

An instance of the TableNode.Cell class.

See Also

EditCellTextEventArgs Members
EditCellTextEventArgs Class
MindFusion.Diagramming Namespace

EditCellTextEventArgs.Column Property

Gets the index of the table's column that contains Cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Column { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Column As Integer
```

Property Value

An integer value specifying the zero-based index of the column.

See Also

EditCellTextEventArgs Members
EditCellTextEventArgs Class
MindFusion.Diagramming Namespace

EditCellTextEventArgs.NewText Property

Gets the text that has just been entered by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string NewText { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NewText As String
```

Property Value

A string specifying the new text of the cell.

See Also

EditCellTextEventArgs Members
EditCellTextEventArgs Class
MindFusion.Diagramming Namespace

EditCellTextEventArgs.OldText Property

Gets the text that had been displayed in the cell before the user edited it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string OldText { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property OldText As String
```

Property Value

A string specifying the old text of the cell.

See Also

EditCellTextEventArgs Members
EditCellTextEventArgs Class
MindFusion.Diagramming Namespace

EditCellTextEventArgs.Row Property

Gets the index of the table's row that contains Cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Row { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Row As Integer
```

Property Value

An integer value specifying the zero-based index of the row.

See Also

EditCellTextEventArgs Members
EditCellTextEventArgs Class
MindFusion.Diagramming Namespace

EditCellTextEventArgs.Table Property

Get the table that contains Cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode Table { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Table As TableNode
```

Property Value

An instance of the TableNode class.

See Also

EditCellTextEventArgs Members
EditCellTextEventArgs Class
MindFusion.Diagramming Namespace

EditLinkTextEventArgs Class

Contains the arguments passed to the EditLinkTextEventHandler event handlers.

Syntax

C# Copy Code

```
public class EditLinkTextEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class EditLinkTextEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.EditLinkTextEventArgs

See Also

EditLinkTextEventArgs Members
MindFusion.Diagramming Namespace

EditLinkTextEventArgs Members

Contains the arguments passed to the EditLinkTextEventHandler event handlers.

The following tables list the members exposed by the EditLinkTextEventArgs type.

Public Constructors

	Name	Description
	EditLinkTextEventArgs	Initializes a new instance of the **EditLinkTextEventArgs** class.

Public Properties

	Name	Description
	Label	Gets or sets a LinkLabel related to this event.
	Link	Gets the link whose text was edited.
	NewText	Gets the new text.
	OldText	Gets the text before in-place editing.

See Also

EditLinkTextEventArgs Class
MindFusion.Diagramming Namespace

EditLinkTextEventArgs Constructor

Initializes a new instance of the EditLinkTextEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public EditLinkTextEventArgs (
    DiagramLink link,
    string oldText,
```

```
    string newText
)
```

Visual Basic Copy Code

```
Public New ( _
    link As DiagramLink, _
    oldText As String, _
    newText As String _
)
```

Parameters

link
The link whose text is edited.

oldText
The old value of the link's Text.

newText
The new value of the link's Text.

See Also

EditLinkTextEventArgs Class
MindFusion.Diagramming Namespace

EditLinkTextEventArgs Properties

EditLinkTextEventArgs.Label Property

Gets or sets a LinkLabel related to this event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkLabel Label { get; set; }
```

Visual Basic Copy Code

```
Public Property Label As LinkLabel
```

Property Value

A LinkLabel reference.

See Also

EditLinkTextEventArgs Members
EditLinkTextEventArgs Class
MindFusion.Diagramming Namespace

EditLinkTextEventArgs.Link Property

Gets the link whose text was edited.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink Link { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Link As DiagramLink
```

Property Value

See Also

EditLinkTextEventArgs Members
EditLinkTextEventArgs Class
MindFusion.Diagramming Namespace

EditLinkTextEventArgs.NewText Property

Gets the new text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string NewText { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NewText As String
```

Property Value

The new value of the link's Text property.

See Also

EditLinkTextEventArgs Members
EditLinkTextEventArgs Class
MindFusion.Diagramming Namespace
OldText Property

EditLinkTextEventArgs.OldText Property

Gets the text before in-place editing.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string OldText { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property OldText As String
```

Property Value

The original value of the Text property of the edited link.

See Also

EditLinkTextEventArgs Members
EditLinkTextEventArgs Class

EditNodeTextEventArgs Class

Contains the arguments passed to the NodeTextEdited event handlers.

Syntax

C# Copy Code

```
public class EditNodeTextEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class EditNodeTextEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.EditNodeTextEventArgs

See Also

EditNodeTextEventArgs Members
MindFusion.Diagramming Namespace

EditNodeTextEventArgs Members

Contains the arguments passed to the NodeTextEdited event handlers.

The following tables list the members exposed by the EditNodeTextEventArgs type.

Public Constructors

	Name	Description
	EditNodeTextEventArgs	Initializes a new instance of the EditNodeTextEventArgs class.

Public Properties

	Name	Description
	Label	Gets or sets a NodeLabel related to this event.
	NewText	Gets the text that has just been entered by the user.
	Node	Gets the node whose text has been edited.
	OldText	Gets the text that had been displayed in the node before the user edited it.

See Also

EditNodeTextEventArgs Class
MindFusion.Diagramming Namespace

EditNodeTextEventArgs Constructor

Initializes a new instance of the EditNodeTextEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public EditNodeTextEventArgs (
    DiagramNode node,
    string oldText,
    string newText
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode, _
    oldText As String, _
    newText As String _
)
```

Parameters

node
The DiagramNode related to the event.

oldText
The old text of the node.

newText
The new text of the node.

See Also

EditNodeTextEventArgs Class
MindFusion.Diagramming Namespace

EditNodeTextEventArgs Properties

EditNodeTextEventArgs.Label Property

Gets or sets a NodeLabel related to this event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeLabel Label { get; set; }
```

Visual Basic Copy Code

```
Public Property Label As NodeLabel
```

Property Value

A NodeLabel instance.

Remarks

This property is set to **null** when the user edits node's main Text. It has a value only if user edits a supplementary node label.

See Also

EditNodeTextEventArgs Members
EditNodeTextEventArgs Class

MindFusion.Diagramming Namespace
DiagramNode.AddLabel Method

EditNodeTextEventArgs.NewText Property

Gets the text that has just been entered by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string NewText { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NewText As String
```

Property Value

A string specifying the new text of the node.

See Also

EditNodeTextEventArgs Members
EditNodeTextEventArgs Class
MindFusion.Diagramming Namespace

EditNodeTextEventArgs.Node Property

Gets the node whose text has been edited.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Node { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Node As DiagramNode
```

Property Value

An instance of the DiagramNode class.

See Also

EditNodeTextEventArgs Members
EditNodeTextEventArgs Class
MindFusion.Diagramming Namespace

EditNodeTextEventArgs.OldText Property

Gets the text that had been displayed in the node before the user edited it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string OldText { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property OldText As String
```

Property Value

A string specifying the old text of the node.

See Also

EditNodeTextEventArgs Members
EditNodeTextEventArgs Class
MindFusion.Diagramming Namespace

EffectsProvider Class

Provides smooth shadow and emboss effects for MvcDiagram nodes.

Syntax

C# Copy Code

```
public sealed class EffectsProvider : IEffectsProvider
```

Visual Basic Copy Code

```
Public NotInheritable Class EffectsProvider
    Implements IEffectsProvider
```

Remarks

You can assign an instance of this class to the EffectsProvider property of the Diagram class.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.EffectsProvider

See Also

EffectsProvider Members
MindFusion.Diagramming Namespace

EffectsProvider Members

Provides smooth shadow and emboss effects for MvcDiagram nodes.

The following tables list the members exposed by the EffectsProvider type.

Public Constructors

	Name	Description
	EffectsProvider	Initializes a new instance of the **EffectsProvider** class.

See Also

EffectsProvider Class
MindFusion.Diagramming Namespace

EffectsProvider Constructor

Initializes a new instance of the EffectsProvider class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming.EffectsProvider

Syntax

C# Copy Code

```
public EffectsProvider ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

EffectsProvider Class
MindFusion.Diagramming Namespace

ElementTemplate Class

Represents an element of a shape definition. This is the parent class of specific element types, such as ArcTemplate, LineTemplate and BezierTemplate.

Syntax

C# Copy Code

```
public abstract class ElementTemplate : ICloneable
```

Visual Basic Copy Code

```
Public MustInherit Class ElementTemplate
    Implements ICloneable
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ElementTemplate
    MindFusion.Diagramming.ArcTemplate
    MindFusion.Diagramming.BezierTemplate
    MindFusion.Diagramming.LineTemplate
    MindFusion.Diagramming.RoundRectangleTemplate

See Also

ElementTemplate Members
MindFusion.Diagramming Namespace
DefaultShape

ElementTemplate Members

Represents an element of a shape definition. This is the parent class of specific element types, such as ArcTemplate, LineTemplate and BezierTemplate.

The following tables list the members exposed by the ElementTemplate type.

Protected Constructors

	Name	Description
	ElementTemplate	Initializes a new instance of the **ElementTemplate** class.

Public Properties

	Name	Description
	Color	Gets or sets the outline color of this element.
	DashStyle	Gets or sets the dash style of the outline of this element.
	Width	Gets or sets the outline width of this element.

Public Methods

	Name	Description
	Clone	Creates an exact copy of this element.
	SameAs	Checks whether the current element and the specified element resemble each other closely enough to be considered identical.

Protected Methods

	Name	Description
	GetElementPen	Gets a pen used to render the element. If no pen properties are explicitly specified for this element, returns the pen passed as an argument.

See Also

ElementTemplate Class
MindFusion.Diagramming Namespace

ElementTemplate Constructor

Initializes a new instance of the ElementTemplate class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected ElementTemplate ()
```

Visual Basic Copy Code

```
Protected Sub New ()
```

See Also

ElementTemplate Class
MindFusion.Diagramming Namespace

ElementTemplate Methods

ElementTemplate.Clone Method

Creates an exact copy of this element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract Object Clone ()
```

Visual Basic Copy Code

```
Public MustOverride Function Clone () As Object
```

Return Value

An instance of the appropriate type that represents the element clone.

See Also

ElementTemplate Members
ElementTemplate Class
MindFusion.Diagramming Namespace

ElementTemplate.GetElementPen Method

Gets a pen used to render the element. If no pen properties are explicitly specified for this element, returns the pen passed as an argument.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected Pen GetElementPen (
    Pen pen
)
```

Visual Basic Copy Code

```
Protected Function GetElementPen( _
    pen As Pen _
) As Pen
```

Parameters

pen
The Pen object specifying the base pen.

Return Value

See Also

ElementTemplate Members
ElementTemplate Class
MindFusion.Diagramming Namespace

ElementTemplate.SameAs Method

Checks whether the current element and the specified element resemble each other closely enough to be considered identical.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract bool SameAs (
    ElementTemplate obj
)
```

Visual Basic — Copy Code

```
Public MustOverride Function SameAs( _
    obj As ElementTemplate _
) As Boolean
```

Parameters

obj
The ElementTemplate to compare to this element.

Return Value

true if the specified object closely approximates this element; otherwise, **false**.

See Also

ElementTemplate Members
ElementTemplate Class
MindFusion.Diagramming Namespace

ElementTemplate Properties

ElementTemplate.Color Property

Gets or sets the outline color of this element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# — Copy Code

```
public Color Color { get; set; }
```

Visual Basic — Copy Code

```
Public Property Color As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** specifying the color of the outline.

See Also

ElementTemplate Members
ElementTemplate Class
MindFusion.Diagramming Namespace

ElementTemplate.DashStyle Property

Gets or sets the dash style of the outline of this element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# — Copy Code

```
public DashStyle DashStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property DashStyle As DashStyle
```

Property Value

A member of the **DashStyle** enumeration specifying the style of the outline.

See Also

ElementTemplate Members
ElementTemplate Class
MindFusion.Diagramming Namespace

ElementTemplate.Width Property

Gets or sets the outline width of this element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Width { get; set; }
```

Visual Basic Copy Code

```
Public Property Width As Double
```

Property Value

A **double** value specifying the width of the outline.

See Also

ElementTemplate Members
ElementTemplate Class
MindFusion.Diagramming Namespace

EmbossEffect Class

Represents an emboss effect.

Syntax

C# Copy Code

```
public sealed class EmbossEffect : NodeEffect
```

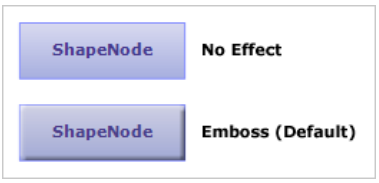
Visual Basic Copy Code

```
Public NotInheritable Class EmbossEffect
    Inherits NodeEffect
```

Remarks

This effect can be applied to the nodes in a Diagram by adding an instance of this class to the NodeEffects collection. In addition, this effect will be active only when the Diagram.EffectsProvider property is set to an object implementing the GenerateEmboss method of the IEffectsProvider interface.

The following image illustrates the Emboss effect applied to a rectangular node:



Example

The following example demonstrates how to apply emboss effect to the nodes of an existing diagram.

C# Copy Code

```
diagram.NodeEffects.Add(new EmbossEffect());
```

Visual Basic Copy Code

```
diagram.NodeEffects.Add(New EmbossEffect())
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.NodeEffect
    MindFusion.Diagramming.EmbossEffect

See Also

EmbossEffect Members
MindFusion.Diagramming Namespace

EmbossEffect Members

Represents an emboss effect.

The following tables list the members exposed by the EmbossEffect type.

Public Constructors

	Name	Description
	EmbossEffect	Initializes a new instance of the **EmbossEffect** class.

Public Methods

	Name	Description
	Clone	NodeEffect.Clone override.

Public Events

	Name	Description
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation. (Inherited from NodeEffect.)

See Also

EmbossEffect Class
MindFusion.Diagramming Namespace

EmbossEffect Constructor

Initializes a new instance of the EmbossEffect class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public EmbossEffect ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

EmbossEffect Class
MindFusion.Diagramming Namespace

EmbossEffect Methods

EmbossEffect.Clone Method

NodeEffect.Clone override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Object Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Object
```

Return Value

The copy of this effect.

See Also

EmbossEffect Members
EmbossEffect Class
MindFusion.Diagramming Namespace

EmbossEffect Events

ExpandButtonAction Enumeration

Specifies what action is triggered by the [±] buttons that are displayed near expandable nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum ExpandButtonAction
```

Visual Basic Copy Code

```
Public Enum ExpandButtonAction
```

Members

	Member name	Description
	ExpandTreeBranch	Expands or collapses the children of a node in a tree-like diagram and raises the TreeExpanded or TreeCollapsed event.
	RaiseEvents	Raises the ExpandButtonClicked event.

Remarks

The members of this enumeration can be assigned to the ExpandButtonAction property of the Diagram class.

See Also

MindFusion.Diagramming Namespace

ExpandButtonPosition Enumeration

Defines values that specify possible locations for expand/collapse buttons.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum ExpandButtonPosition
```

Visual Basic Copy Code

```
Public Enum ExpandButtonPosition
```

Members

	Member name	Description
	OuterRight	Place expand/collapse buttons at the right-hand side of their nodes.
	OuterLowerRight	Place expand/collapse buttons at the lower-right corner of their nodes.
	OuterUpperRight	Place expand/collapse buttons at the upper-right corner of their nodes.
	OuterBottom	Place expand/collapse buttons at the bottom side of their nodes.
	OuterLeft	Place expand/collapse buttons at the left-hand side of their nodes.
	OuterLowerLeft	Place expand/collapse buttons at the lower-left corner of their nodes.
	OuterUpperLeft	Place expand/collapse buttons at the upper-left corner of their nodes.
	OuterTop	Place expand/collapse buttons at the top side of their nodes.

Remarks

Assign values from this enumeration to the ExpandButtonPosition property of the Diagram class.

See Also

MindFusion.Diagramming Namespace

Factory Class

Provides shortcut methods for creating and adding instances of the standard MvcDiagram item types. Call the class methods through the Factory property of the Diagram class.

Syntax

C# Copy Code

```
public class Factory
```

Visual Basic Copy Code

```
Public Class Factory
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Factory

See Also

Factory Members
MindFusion.Diagramming Namespace

Factory Members

Provides shortcut methods for creating and adding instances of the standard MvcDiagram item types. Call the class methods through the Factory property of the Diagram class.

The following tables list the members exposed by the Factory type.

Public Constructors

	Name	Description
	Factory	Initializes a new instance of the **Factory** class over the specified **Diagram**.

Public Methods

	Name	Description
	CreateContainerNode	Overloaded. Creates a new ContainerNode instance and adds it to the Nodes collection of the underlying diagram.
	CreateDiagramLink	Overloaded. Creates a new DiagramLink instance and adds it to the Links collection of the underlying diagram.
	CreateGroup	Creates a new hierarchical Group with the specified main item.
	CreatePagedContainerNode	Overloaded.
	CreateShapeNode	Overloaded. Creates a new ShapeNode instance and adds it to the **Nodes** collection of the underlying diagram.
	CreateSvgNode	Overloaded. Creates a new SvgNode instance.
	CreateTableNode	Overloaded. Creates a new TableNode instance and adds it to the **Nodes** collection of the underlying diagram.
	CreateTreeViewNode	Overloaded. Creates a new TreeViewNode instance.

See Also

Factory Class
MindFusion.Diagramming Namespace

Factory Constructor

Initializes a new instance of the Factory class over the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Factory (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram _
)
```

Parameters

diagram
The Diagram object for which to create the new factory.

See Also

Factory Class
MindFusion.Diagramming Namespace

Factory Methods

Factory.CreateContainerNode Method

Creates a new ContainerNode instance and adds it to the Nodes collection of the underlying diagram.

Overload List

Name	Description
Factory.CreateContainerNode (Double, Double, Double, Double)	Creates a new **ContainerNode** instance at the specified location with the specified size and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateContainerNode (Double, Double, Double, Double, Boolean)	Creates a new **ContainerNode** instance at the specified location with the specified size, optionally foldable, and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateContainerNode (Point, Size)	Creates a new **ContainerNode** instance at the specified location with the specified size and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateContainerNode (Point, Size, Boolean)	Creates a new **ContainerNode** instance at the specified location with the specified size, optionally foldable, and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateContainerNode (Rect)	Creates a new **ContainerNode** instance with the specified dimensions and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateContainerNode (Rect, Boolean)	Creates a new **ContainerNode** instance with the specified dimensions, optionally foldable, and adds it to the **Nodes** collection of the underlying diagram.

See Also

Factory.CreateContainerNode Method (Double, Double, Double, Double)

Creates a new ContainerNode instance at the specified location with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNode CreateContainerNode (
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Public Function CreateContainerNode( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
) As ContainerNode
```

Parameters

x
The x-coordinate of the upper-left corner of the new node.

y
The y-coordinate of the upper-left corner of the new node.

width
The width of the new node.

height
The height of the new node.

Return Value

The newly created ContainerNode instance.

Remarks

The method creates a container node at the position specified by x and y. The size of the container is set by *width* and *height*. For further customization of the node, use the properties and methods of the ContainerNode class.

See Also

CreateContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateDiagramLink Method
CreateTableNode Method
Diagram.Nodes Property
Diagram.AlignPointToGrid Method

Factory.CreateContainerNode Method (Double, Double, Double, Double, Boolean)

Creates a new ContainerNode instance at the specified location with the specified size, optionally foldable, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNode CreateContainerNode (
    double x,
    double y,
    double width,
    double height,
    bool foldable
)
```

Visual Basic Copy Code

```
Public Function CreateContainerNode( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double, _
    foldable As Boolean _
) As ContainerNode
```

Parameters

x
The x-coordinate of the upper-left corner of the new node.

y
The y-coordinate of the upper-left corner of the new node.

width
The width of the new node.

height
The height of the new node.

foldable
true if the container can be folded; otherwise, **false**.

Return Value

The newly created ContainerNode instance.

Remarks

The method creates a container node at the position specified by *x* and *y*. The size of the container is set by *width* and *height*. The parameter *foldable* specifies whether the container node is initially foldable. For further customization of the node, use the properties and methods of the ContainerNode class.

See Also

CreateContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateContainerNode Method (Point, Size)

Creates a new ContainerNode instance at the specified location with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNode CreateContainerNode (
    Point location,
    Size size
)
```

Visual Basic Copy Code

```
Public Function CreateContainerNode( _
    location As Point, _
    size As Size _
) As ContainerNode
```

Parameters

location

The location of the ContainerNode object.

size

The size of the **ContainerNode** object.

Return Value

The newly created **ContainerNode** instance.

Remarks

The method creates a container node at the position specified by *location*. The size of the container is set by *size*. For further customization of the node, use the properties and methods of the ContainerNode class.

See Also

CreateContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateDiagramLink Method
CreateTableNode Method
Diagram.Nodes Property
Diagram.AlignPointToGrid Method

Factory.CreateContainerNode Method (Point, Size, Boolean)

Creates a new ContainerNode instance at the specified location with the specified size, optionally foldable, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNode CreateContainerNode (
    Point location,
    Size size,
    bool foldable
)
```

Visual Basic Copy Code

```
Public Function CreateContainerNode( _
    location As Point, _
    size As Size, _
    foldable As Boolean _
) As ContainerNode
```

Parameters

location
The location of the ContainerNode object.

size
The newly created **ContainerNode** instance.

foldable
true if the container can be folded; otherwise, **false**.

Return Value

The newly created **ContainerNode** instance.

Remarks

The method creates a container node at the position specified by *location*. The size of the container is set by *size*. The parameter *foldable* specifies whether the container node is initially foldable. For further customization of the node, use the properties and methods of the ContainerNode class.

See Also

CreateContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateDiagramLink Method
CreateTableNode Method
Diagram.Nodes Property
Diagram.AlignPointToGrid Method

Factory.CreateContainerNode Method (Rect)

Creates a new ContainerNode instance with the specified dimensions and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNode CreateContainerNode (
    Rect bounds
)
```

Visual Basic Copy Code

```
Public Function CreateContainerNode( _
    bounds As Rect _
) As ContainerNode
```

Parameters

bounds
The initial bounding rectangle of the new node.

Return Value

The newly created ContainerNode instance.

Remarks

The method creates a container node with the position and size specified by *bounds*. For further customization of the node, use the properties and methods of the ContainerNode class.

See Also

CreateContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateContainerNode Method (Rect, Boolean)

Creates a new ContainerNode instance with the specified dimensions, optionally foldable, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerNode CreateContainerNode (
    Rect bounds,
    bool foldable
)
```

Visual Basic Copy Code

```
Public Function CreateContainerNode( _
    bounds As Rect, _
    foldable As Boolean _
) As ContainerNode
```

Parameters

bounds
The initial bounding rectangle of the new node.

foldable
true if the container can be folded; otherwise, **false**.

Return Value

The newly created ContainerNode instance.

Remarks

The method creates a container node with the position and size specified by *bounds*. The parameter *foldable* specifies whether the container node is initially foldable. For further customization of the node, use the properties and methods of the ContainerNode class.

See Also

CreateContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateDiagramLink Method

Creates a new DiagramLink instance and adds it to the Links collection of the underlying diagram.

Overload List

Name	Description
Factory.CreateDiagramLink (DiagramNode, DiagramNode)	Creates a new **DiagramLink** instance between the specified nodes and adds it to the **Links** collection of the underlying diagram.
Factory.CreateDiagramLink (DiagramNode, Point)	Creates a new **DiagramLink** instance between the specified node and the specified point, and adds it to the **Links** collection of the underlying diagram.
Factory.CreateDiagramLink (DiagramNode, TableNode, Int32)	Creates a new **DiagramLink** instance between the specified nodes and the specified table, and adds it to the **Links** collection of the underlying diagram.
Factory.CreateDiagramLink (Point, DiagramNode)	Creates a new **DiagramLink** instance between the specified point and the specified node, and adds it to the **Links** collection of the underlying diagram.

Factory.CreateDiagramLink (Point, Point)	Creates a new **DiagramLink** instance between the specified points and adds it to the **Links** collection of the underlying diagram.
Factory.CreateDiagramLink (ShapeNode, Int32, ShapeNode, Int32)	Creates a new **DiagramLink** instance between the specified nodes and adds it to the **Links** collection of the underlying diagram.
Factory.CreateDiagramLink (TableNode, Int32, DiagramNode)	Creates a new **DiagramLink** instance between the specified table and the specified node, and adds it to the **Links** collection of the underlying diagram.
Factory.CreateDiagramLink (TableNode, Int32, TableNode, Int32)	Creates a new **DiagramLink** instance between the specified tables and adds it to the **Links** collection of the underlying diagram.
Factory.CreateDiagramLink (TableNode, TableNode)	Creates a new **DiagramLink** instance between the specified tables and adds it to the **Links** collection of the underlying diagram.
Factory.CreateDiagramLink (TreeViewNode, TreeViewItem, TreeViewNode, TreeViewItem)	Creates a new **DiagramLink** instance between the specified tree items and adds it to the **Links** collection of the underlying diagram.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateDiagramLink Method (DiagramNode, DiagramNode)

Creates a new DiagramLink instance between the specified nodes and adds it to the Links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink CreateDiagramLink (
    DiagramNode origin,
    DiagramNode destination
)
```

Visual Basic Copy Code

```
Public Function CreateDiagramLink( _
    origin As DiagramNode, _
    destination As DiagramNode _
) As DiagramLink
```

Parameters

origin
The origin node.

destination
The destination node.

Return Value

The newly created DiagramLink instance.

Remarks

The method creates a link connecting the nodes specified by *origin* and *destination*. For further customization of the link, use the properties and methods of the DiagramLink class.

See Also

CreateDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateTableNode Method

CreateShapeNode Method
Diagram.Links Property

Factory.CreateDiagramLink Method (DiagramNode, Point)

Creates a new DiagramLink instance between the specified node and the specified point, and adds it to the Links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink CreateDiagramLink (
    DiagramNode origin,
    Point destination
)
```

Visual Basic Copy Code

```
Public Function CreateDiagramLink( _
    origin As DiagramNode, _
    destination As Point _
) As DiagramLink
```

Parameters

origin
The origin node.

destination
The destination point.

Return Value

The newly created DiagramLink instance.

Remarks

The method creates a link connecting the node specified by *origin* and the point specified by *destination*. For further customization of the link, use the properties and methods of the DiagramLink class.

See Also

CreateDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateTableNode Method
Diagram.Links Property

Factory.CreateDiagramLink Method (DiagramNode, TableNode, Int32)

Creates a new DiagramLink instance between the specified nodes and the specified table, and adds it to the Links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink CreateDiagramLink (
    DiagramNode origin,
```

```
    TableNode destination,
    int destRow
)
```

Visual Basic Copy Code

```
Public Function CreateDiagramLink( _
    origin As DiagramNode, _
    destination As TableNode, _
    destRow As Integer _
) As DiagramLink
```

Parameters

origin
The origin node.

destination
The destination node.

destRow
The zero-based index of the destination table row of the new link.

Return Value

The newly created DiagramLink instance.

Remarks

The method creates a link connecting the node specified by *origin* and the table specified by *destination*. For further customization of the link, use the properties and methods of the DiagramLink class.

See Also

CreateDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateTableNode Method
Diagram.Links Property

Factory.CreateDiagramLink Method (Point, DiagramNode)

Creates a new DiagramLink instance between the specified point and the specified node, and adds it to the Links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink CreateDiagramLink (
    Point origin,
    DiagramNode destination
)
```

Visual Basic Copy Code

```
Public Function CreateDiagramLink( _
    origin As Point, _
    destination As DiagramNode _
) As DiagramLink
```

Parameters

origin
The origin point.

destination
The destination node.

Return Value

The newly created DiagramLink instance.

Remarks

The method creates a link connecting the point specified by *origin* and the node specified by *destination*. For further customization of the link, use the properties and methods of the DiagramLink class.

See Also

CreateDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateTableNode Method
Diagram.Links Property

Factory.CreateDiagramLink Method (Point, Point)

Creates a new DiagramLink instance between the specified points and adds it to the Links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink CreateDiagramLink (
    Point origin,
    Point destination
)
```

Visual Basic Copy Code

```
Public Function CreateDiagramLink( _
    origin As Point, _
    destination As Point _
) As DiagramLink
```

Parameters

origin
The origin point.

destination
The destination point.

Return Value

The newly created DiagramLink instance.

Remarks

The method creates a link connecting the points specified by *origin* and *destination*. For further customization of the link, use the properties and methods of the DiagramLink class.

See Also

CreateDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateTableNode Method
Diagram.Links Property

Factory.CreateDiagramLink Method (ShapeNode, Int32, ShapeNode, Int32)

Creates a new DiagramLink instance between the specified nodes and adds it to the Links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink CreateDiagramLink (
    ShapeNode origin,
    int originAnchor,
    ShapeNode destination,
    int destAnchor
)
```

Visual Basic Copy Code

```
Public Function CreateDiagramLink( _
    origin As ShapeNode, _
    originAnchor As Integer, _
    destination As ShapeNode, _
    destAnchor As Integer _
) As DiagramLink
```

Parameters

origin
The origin node.

originAnchor
The index of an anchor point within the origin node's AnchorPattern.

destination
The destination node.

destAnchor
The index of an anchor point within the destination node's **AnchorPattern**.

Return Value

The newly created DiagramLink instance.

Remarks

The method creates a link connecting the shape nodes specified by *origin* and *destination*. For further customization of the link, use the properties and methods of the DiagramLink class.

See Also

CreateDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateTableNode Method
Links Property

Factory.CreateDiagramLink Method (TableNode, Int32, DiagramNode)

Creates a new DiagramLink instance between the specified table and the specified node, and adds it to the Links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink CreateDiagramLink (
    TableNode origin,
    int originRow,
    DiagramNode destination
)
```

Visual Basic Copy Code

```
Public Function CreateDiagramLink( _
    origin As TableNode, _
    originRow As Integer, _
    destination As DiagramNode _
) As DiagramLink
```

Parameters

origin
The origin node.

originRow
The zero-based index of the destination table row of the new link.

destination
The destination node.

Return Value

The newly created DiagramLink instance.

Remarks

The method creates a link connecting the table specified by *origin* and the node specified by *destination*. For further customization of the link, use the properties and methods of the DiagramLink class.

See Also

CreateDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateTableNode Method
Diagram.Links Property

Factory.CreateDiagramLink Method (TableNode, Int32, TableNode, Int32)

Creates a new DiagramLink instance between the specified tables and adds it to the Links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink CreateDiagramLink (
    TableNode origin,
    int originRow,
    TableNode destination,
    int destRow
)
```

Visual Basic Copy Code

```
Public Function CreateDiagramLink( _
    origin As TableNode, _
```

```
    originRow As Integer, _
    destination As TableNode, _
    destRow As Integer _
) As DiagramLink
```

Parameters

origin
The origin node.

originRow
The zero-based index of the origin table row of the new link.

destination
The destination node.

destRow
The zero-based index of the destination table row of the new link.

Return Value

The newly created DiagramLink instance.

Remarks

The method creates a link connecting the tables specified by *origin* and *destination*. For further customization of the link, use the properties and methods of the DiagramLink class.

See Also

CreateDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateTableNode Method
Diagram.Links Property

Factory.CreateDiagramLink Method (TableNode, TableNode)

Creates a new DiagramLink instance between the specified tables and adds it to the Links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink CreateDiagramLink (
    TableNode origin,
    TableNode destination
)
```

Visual Basic Copy Code

```
Public Function CreateDiagramLink( _
    origin As TableNode, _
    destination As TableNode _
) As DiagramLink
```

Parameters

origin
The origin node.

destination
The destination node.

Return Value

The newly created DiagramLink instance.

Remarks

The method creates a link connecting the tables specified by *origin* and *destination*. For further customization of the link, use the properties and methods of the DiagramLink class.

See Also

CreateDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateTableNode Method
Diagram.Links Property

Factory.CreateDiagramLink Method (TreeViewNode, TreeViewItem, TreeViewNode, TreeViewItem)

Creates a new DiagramLink instance between the specified tree items and adds it to the Links collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink CreateDiagramLink (
    TreeViewNode origin,
    TreeViewItem originItem,
    TreeViewNode destination,
    TreeViewItem destinationItem
)
```

Visual Basic Copy Code

```
Public Function CreateDiagramLink( _
    origin As TreeViewNode, _
    originItem As TreeViewItem, _
    destination As TreeViewNode, _
    destinationItem As TreeViewItem _
) As DiagramLink
```

Parameters

origin
The origin node.

originItem
The origin item.

destination
The destination node.

destinationItem
The destination item.

Return Value

The newly created DiagramLink instance.

See Also

CreateDiagramLink Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateGroup Method

Creates a new hierarchical Group with the specified main item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Group CreateGroup (
    DiagramItem mainItem
)
```

Visual Basic Copy Code

```
Public Function CreateGroup( _
    mainItem As DiagramItem _
) As Group
```

Parameters

mainItem
The DiagramItem around which the group would be built.

Return Value

The newly created Group instance if successfully created; otherwise **null** (**Nothing** in Visual Basic).

Remarks

Use this method to create an hierarchical group of items. Attach subordinated items by using the various Group members.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace
Diagram.GroupDestroyed Event

Factory.CreatePagedContainerNode Method

Overload List

Name	Description
Factory.CreatePagedContainerNode (Double, Double, Double, Double)	Creates a new PagedContainerNode instance at the specified location with the specified size and adds it to the Nodes collection of the underlying diagram.
Factory.CreatePagedContainerNode (Double, Double, Double, Double, IEnumerable(String))	Creates a new PagedContainerNode instance at the specified location with the specified size and page titles, and adds it to the Nodes collection of the underlying diagram.
Factory.CreatePagedContainerNode (Point, Size)	Creates a new PagedContainerNode instance at the specified location with the specified size and adds it to the Nodes collection of the underlying diagram.
Factory.CreatePagedContainerNode (Point, Size, IEnumerable(String))	Creates a new PagedContainerNode instance at the specified location with the specified size and page titles, and adds it to the Nodes collection of the underlying diagram.
Factory.CreatePagedContainerNode (Rect)	Creates a new PagedContainerNode instance with the specified dimensions and adds it to the Nodes collection of the underlying diagram.
Factory.CreatePagedContainerNode (Rect, IEnumerable(String))	Creates a new PagedContainerNode instance with the specified dimensions and page titles, and adds it to the Nodes collection of the underlying diagram.

See Also

Factory Members
Factory Class

Factory.CreatePagedContainerNode Method (Double, Double, Double, Double)

Creates a new PagedContainerNode instance at the specified location with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PagedContainerNode CreatePagedContainerNode (
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Public Function CreatePagedContainerNode( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
) As PagedContainerNode
```

Parameters

x
The x-coordinate of the upper-left corner of the new node.

y
The y-coordinate of the upper-left corner of the new node.

width
The width of the new node.

height
The height of the new node.

Return Value

The newly created PagedContainerNode instance.

Remarks

The method creates a paged container node at the position specified by x and y. The size of the container is set by width and height.

See Also

CreatePagedContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreatePagedContainerNode Method (Double, Double, Double, Double, IEnumerable(String))

Creates a new PagedContainerNode instance at the specified location with the specified size and page titles, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PagedContainerNode CreatePagedContainerNode (
    double x,
    double y,
    double width,
    double height,
    IEnumerable<string> pageTitles
)
```

Visual Basic Copy Code

```
Public Function CreatePagedContainerNode( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double, _
    pageTitles As IEnumerable(Of String) _
) As PagedContainerNode
```

Parameters

x
The x-coordinate of the upper-left corner of the new node.

y
The y-coordinate of the upper-left corner of the new node.

width
The width of the new node.

height
The height of the new node.

pageTitles
A list of strings specifying the titles of the initial pages.

Return Value

The newly created PagedContainerNode instance.

Remarks

The method creates a paged container node at the position specified by x and y. The size of the container is set by width and height. The parameter pageTitles specifies the titles of the initial pages of the container.

See Also

CreatePagedContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreatePagedContainerNode Method (Point, Size)

Creates a new PagedContainerNode instance at the specified location with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PagedContainerNode CreatePagedContainerNode (
    Point location,
    Size size
)
```

Visual Basic Copy Code

```
Public Function CreatePagedContainerNode( _
    location As Point, _
    size As Size _
) As PagedContainerNode
```

Parameters

location
The location of the PagedContainerNode object.

size
The size of the PagedContainerNode object.

Return Value

The newly created PagedContainerNode instance.

Remarks

The method creates a paged container node at the position specified by location. The size of the container is set by size.

See Also

CreatePagedContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreatePagedContainerNode Method (Point, Size, IEnumerable(String))

Creates a new PagedContainerNode instance at the specified location with the specified size and page titles, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PagedContainerNode CreatePagedContainerNode (
    Point location,
    Size size,
    IEnumerable<string> pageTitles
)
```

Visual Basic Copy Code

```
Public Function CreatePagedContainerNode( _
    location As Point, _
    size As Size, _
    pageTitles As IEnumerable(Of String) _
) As PagedContainerNode
```

Parameters

location
The location of the PagedContainerNode object.

size
The size of the PagedContainerNode object.

pageTitles
A list of strings specifying the titles of the initial pages.

Return Value

The newly created PagedContainerNode instance.

Remarks

The method creates a paged container node at the position specified by location. The size of the container is set by size. The parameter pageTitles specifies the titles of the initial pages of the container.

See Also

CreatePagedContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreatePagedContainerNode Method (Rect)

Creates a new PagedContainerNode instance with the specified dimensions and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PagedContainerNode CreatePagedContainerNode (
    Rect bounds
)
```

Visual Basic Copy Code

```
Public Function CreatePagedContainerNode( _
    bounds As Rect _
) As PagedContainerNode
```

Parameters

bounds
The initial bounding rectangle of the new node.

Return Value

The newly created PagedContainerNode instance.

Remarks

The method creates a paged container node with the position and size specified by bounds.

See Also

CreatePagedContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreatePagedContainerNode Method (Rect, IEnumerable(String))

Creates a new PagedContainerNode instance with the specified dimensions and page titles, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PagedContainerNode CreatePagedContainerNode (
    Rect bounds,
    IEnumerable<string> pageTitles
)
```

Visual Basic Copy Code

```
Public Function CreatePagedContainerNode( _
    bounds As Rect, _
    pageTitles As IEnumerable(Of String) _
) As PagedContainerNode
```

Parameters

bounds
The initial bounding rectangle of the new node.

pageTitles
A list of strings specifying the titles of the initial pages.

Return Value

The newly created PagedContainerNode instance.

Remarks

The method creates a paged container node with the position and size specified by bounds. The parameter pageTitles specifies the titles of the initial pages of the container.

See Also

CreatePagedContainerNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateShapeNode Method

Creates a new ShapeNode instance and adds it to the Nodes collection of the underlying diagram.

Overload List

Name	Description
Factory.CreateShapeNode (Double, Double, Double, Double)	Creates a new **ShapeNode** instance at the specified position with the specified size and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateShapeNode (Double, Double, Double, Double, Shape)	Creates a new **ShapeNode** instance at the specified position with the specified size and shape, and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateShapeNode (Point, Size)	Creates a new **ShapeNode** instance at the specified position with the specified size and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateShapeNode (Point, Size, Shape)	Creates a new **ShapeNode** instance at the specified position with the specified size and shape, and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateShapeNode (Rect)	Creates a new **ShapeNode** instance with the specified dimensions and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateShapeNode (Rect, Shape)	Creates a new **ShapeNode** instance with the specified dimensions and shape, and adds it to the **Nodes** collection of the underlying diagram.

See Also

Factory.CreateShapeNode Method (Double, Double, Double, Double)

Creates a new ShapeNode instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNode CreateShapeNode (
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Public Function CreateShapeNode( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
) As ShapeNode
```

Parameters

x
The x-coordinate of the upper-left corner of the new node.

y
The y-coordinate of the upper-left corner of the new node.

width
The width of the new node.

height
The height of the new node.

Return Value

The newly created ShapeNode instance.

Remarks

The method creates a shape node at the position specified by *x* and *y*. The size of the node is set by *width* and *height*. For further customization of the node, use the properties and methods of the ShapeNode class.

See Also

CreateShapeNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateDiagramLink Method
CreateTableNode Method
Diagram.Nodes Property
Diagram.AlignPointToGrid Method

Factory.CreateShapeNode Method (Double, Double, Double, Double, Shape)

Creates a new ShapeNode instance at the specified position with the specified size and shape, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNode CreateShapeNode (
    double x,
    double y,
    double width,
    double height,
    Shape shape
)
```

Visual Basic Copy Code

```
Public Function CreateShapeNode( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double, _
    shape As Shape _
) As ShapeNode
```

Parameters

x
The x-coordinate of the upper-left corner of the new node.

y
The y-coordinate of the upper-left corner of the new node.

width
The width of the new node.

height
The height of the new node.

shape
The shape of the new node.

Return Value

The newly created ShapeNode instance.

Remarks

The method creates a shape node at the position specified by *x* and *y*. The size of the node is set by *width* and *height*. The shape of the new node is specified by *shape*. For further customization of the node, use the properties and methods of the ShapeNode class.

See Also

CreateShapeNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateDiagramLink Method
CreateTableNode Method
Diagram.Nodes Property
Diagram.AlignPointToGrid Method

Factory.CreateShapeNode Method (Point, Size)

Creates a new ShapeNode instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNode CreateShapeNode (
    Point location,
    Size size
)
```

Visual Basic Copy Code

```
Public Function CreateShapeNode( _
    location As Point, _
    size As Size _
) As ShapeNode
```

Parameters

location

The position of the new node.

size

The size of the new node.

Return Value

The newly created ShapeNode instance.

Remarks

The method creates a shape node at the position specified by *location*. The size of the node is set by *size*. For further customization of the node, use the properties and methods of the ShapeNode class.

See Also

CreateShapeNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateDiagramLink Method
CreateTableNode Method
Diagram.Nodes Property
Diagram.AlignPointToGrid Method

Factory.CreateShapeNode Method (Point, Size, Shape)

Creates a new ShapeNode instance at the specified position with the specified size and shape, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNode CreateShapeNode (
    Point location,
    Size size,
    Shape shape
)
```

Visual Basic Copy Code

```
Public Function CreateShapeNode( _
    location As Point, _
    size As Size, _
```

```
    shape As Shape _
) As ShapeNode
```

Parameters

location
The position of the new node.

size
The size of the new node.

shape
The shape of the new node.

Return Value

The newly created ShapeNode instance.

Remarks

The method creates a shape node at the position specified by *location*. The size of the node is set by *size*. The shape of the new node is specified by *shape*. For further customization of the node, use the properties and methods of the ShapeNode class.

See Also

CreateShapeNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateDiagramLink Method
CreateTableNode Method
Diagram.Nodes Property
Diagram.AlignPointToGrid Method

Factory.CreateShapeNode Method (Rect)

Creates a new ShapeNode instance with the specified dimensions and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNode CreateShapeNode (
    Rect bounds
)
```

Visual Basic Copy Code

```
Public Function CreateShapeNode( _
    bounds As Rect _
) As ShapeNode
```

Parameters

bounds
The bounding rectangle of the new node.

Return Value

The newly created ShapeNode instance.

Remarks

The method creates a shape node with dimensions specified by *bounds*. For further customization of the node, use the properties and methods of the ShapeNode class.

See Also

CreateShapeNode Method Overload List
Factory Members
Factory Class

MindFusion.Diagramming Namespace
CreateDiagramLink Method
CreateTableNode Method
Diagram.Nodes Property
Diagram.AlignPointToGrid Method

Factory.CreateShapeNode Method (Rect, Shape)

Creates a new ShapeNode instance with the specified dimensions and shape, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNode CreateShapeNode (
    Rect bounds,
    Shape shape
)
```

Visual Basic Copy Code

```
Public Function CreateShapeNode( _
    bounds As Rect, _
    shape As Shape _
) As ShapeNode
```

Parameters

bounds
The bounding rectangle of the new node.

shape
The shape of the new node.

Return Value

The newly created ShapeNode instance.

Remarks

The method creates a shape node with dimensions specified by *bounds*. The shape of the new node is specified by *shape*. For further customization of the node, use the properties and methods of the ShapeNode class.

See Also

CreateShapeNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateDiagramLink Method
CreateTableNode Method
Diagram.Nodes Property
Diagram.AlignPointToGrid Method

Factory.CreateSvgNode Method

Creates a new SvgNode instance.

Overload List

Name	Description
Factory.CreateSvgNode (Double, Double, Double, Double)	Creates a new **SvgNode** instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.

Factory.CreateSvgNode (Double, Double, Double, Double, SvgContent)	Creates a new **SvgNode** instance at the specified position with the specified size and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateSvgNode (Point, Size)	Creates a new **SvgNode** instance at the specified position with the specified size and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateSvgNode (Point, Size, SvgContent)	Creates a new **SvgNode** instance at the specified position with the specified size and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateSvgNode (Rect)	Creates a new **SvgNode** instance with the specified dimensions and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateSvgNode (Rect, SvgContent)	Creates a new **SvgNode** instance with the specified dimensions and adds it to the **Nodes** collection of the underlying diagram.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateSvgNode Method (Double, Double, Double, Double)

Creates a new SvgNode instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNode CreateSvgNode (
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Public Function CreateSvgNode( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
) As SvgNode
```

Parameters

x
The x-coordinate of the upper-left corner of the new node.

y
The y-coordinate of the upper-left corner of the new node.

width
The width of the new node.

height
The height of the new node.

Return Value

The newly created SvgNode instance.

Remarks

The method creates an SVG node at the position specified by *x* and *y*. The size of the node is set by *width* and *height*.

See Also

CreateSvgNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateSvgNode Method (Double, Double, Double, Double, SvgContent)

Creates a new SvgNode instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNode CreateSvgNode (
    double x,
    double y,
    double width,
    double height,
    SvgContent content
)
```

Visual Basic Copy Code

```
Public Function CreateSvgNode( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double, _
    content As SvgContent _
) As SvgNode
```

Parameters

x
The x-coordinate of the upper-left corner of the new node.

y
The y-coordinate of the upper-left corner of the new node.

width
The width of the new node.

height
The height of the new node.

content
The content of the new node.

Return Value

The newly created SvgNode instance.

Remarks

The method creates an SVG node at the position specified by *x* and *y*. The size of the node is set by *width* and *height*.

See Also

CreateSvgNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateSvgNode Method (Point, Size)

Creates a new SvgNode instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNode CreateSvgNode (
    Point location,
    Size size
)
```

Visual Basic Copy Code

```
Public Function CreateSvgNode( _
    location As Point, _
    size As Size _
) As SvgNode
```

Parameters

location

The position of the new node.

size

The size of the new node.

Return Value

The newly created SvgNode instance.

Remarks

The method creates an SVG node at the position specified by *location*. The size of the node is set by *size*.

See Also

CreateSvgNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateSvgNode Method (Point, Size, SvgContent)

Creates a new SvgNode instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNode CreateSvgNode (
    Point location,
    Size size,
    SvgContent content
)
```

Visual Basic Copy Code

```
Public Function CreateSvgNode( _
    location As Point, _
    size As Size, _
```

```
    content As SvgContent _
) As SvgNode
```

Parameters

location
The position of the new node.

size
The size of the new node.

content
The content of the new node.

Return Value

The newly created SvgNode instance.

Remarks

The method creates an SVG node at the position specified by *location*. The size of the node is set by *size*.

Example

The following example illustrates how to create new **SvgContent** from an SVG file and how to create an SvgNode based on this content. The variable *diagram* references an existing Diagram object and *fileName* contains the name of the SVG file to load.

C# Copy Code

```
SvgNode node = diagram.Factory.CreateSvgNode(0, 0, 20, 20);

SvgContent content = new SvgContent();
content.Parse(fileName);

node.Content = content;
```

Visual Basic Copy Code

```
Dim node As SvgNode = diagram.Factory.CreateSvgNode(0, 0, 20, 20)

Dim content As New SvgContent()
content.Parse(fileName)

node.Content = content
```

See Also

CreateSvgNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateSvgNode Method (Rect)

Creates a new SvgNode instance with the specified dimensions and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNode CreateSvgNode (
    Rect bounds
)
```

Visual Basic Copy Code

```
Public Function CreateSvgNode( _
    bounds As Rect _
```

```
) As SvgNode
```

Parameters

bounds
The bounding rectangle of the new node.

Return Value

The newly created SvgNode instance.

Remarks

The method creates an SVG node with dimensions specified by *bounds*.

See Also

CreateSvgNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateSvgNode Method (Rect, SvgContent)

Creates a new SvgNode instance with the specified dimensions and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNode CreateSvgNode (
    Rect bounds,
    SvgContent content
)
```

Visual Basic Copy Code

```
Public Function CreateSvgNode( _
    bounds As Rect, _
    content As SvgContent _
) As SvgNode
```

Parameters

bounds
The bounding rectangle of the new node.

content
The content of the new node.

Return Value

The newly created SvgNode instance.

Remarks

The method creates an SVG node with dimensions specified by *bounds*.

See Also

CreateSvgNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateTableNode Method

Creates a new TableNode instance and adds it to the Nodes collection of the underlying diagram.

Overload List

Name	Description
Factory.CreateTableNode (Double, Double, Double, Double)	Creates a new **TableNode** instance at the specified position with the specified size and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateTableNode (Double, Double, Double, Double, Int32, Int32)	Creates a new **TableNode** instance at the specified position with the specified size and the specified number of columns and rows, and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateTableNode (Point, Size)	Creates a new **TableNode** instance at the specified position with the specified size and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateTableNode (Point, Size, Int32, Int32)	Creates a new **TableNode** instance at the specified position with the specified size and with the specified number of columns and rows, and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateTableNode (Rect)	Creates a new **TableNode** instance with the specified dimensions and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateTableNode (Rect, Int32, Int32)	Creates a new **TableNode** instance with the specified dimensions and with the specified number of columns and rows, and adds it to the **Nodes** collection of the underlying diagram.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateTableNode Method (Double, Double, Double, Double)

Creates a new TableNode instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode CreateTableNode (
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Public Function CreateTableNode( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
) As TableNode
```

Parameters

x
The x-coordinate of the upper-left corner of the new table.

y
The y-coordinate of the upper-left corner of the new table.

width
The width of the new table.

height
The height of the new table.

Return Value

The newly created TableNode instance.

Remarks

The method creates a table node at the position specified by *x* and *y*. The size of the node is set by *width* and *height*. For further customization of the node, use the properties and methods of the TableNode class.

See Also

CreateTableNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateDiagramLink Method
Diagram.AlignPointToGrid Method

Factory.CreateTableNode Method (Double, Double, Double, Double, Int32, Int32)

Creates a new TableNode instance at the specified position with the specified size and the specified number of columns and rows, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode CreateTableNode (
    double x,
    double y,
    double width,
    double height,
    int columns,
    int rows
)
```

Visual Basic Copy Code

```
Public Function CreateTableNode( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double, _
    columns As Integer, _
    rows As Integer _
) As TableNode
```

Parameters

x
The x-coordinate of the upper-left corner of the new table.

y
The y-coordinate of the upper-left corner of the new table.

width

The width if the new table.

height
The height of the new table.

columns
The initial number of columns in the table.

rows
The initial number of rows in the table.

Return Value

The newly created TableNode instance.

Remarks

The method creates a table node at the position specified by *x* and *y*. The size of the node is set by *width* and *height*. The initial number of columns and rows in the table is specified by *columns* and *rows* respectively. For further customization of the node, use the properties and methods of the TableNode class.

See Also

CreateTableNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateDiagramLink Method
Diagram.AlignPointToGrid Method

Factory.CreateTableNode Method (Point, Size)

Creates a new TableNode instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode CreateTableNode (
    Point location,
    Size size
)
```

Visual Basic Copy Code

```
Public Function CreateTableNode( _
    location As Point, _
    size As Size _
) As TableNode
```

Parameters

location
The position of the new table.

size
The size of the new table.

Return Value

The newly created TableNode instance.

Remarks

The method creates a table node at the position specified by *location*. The size of the node is set by *size*. For further customization of the node, use the properties and methods of the TableNode class.

See Also

CreateTableNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateDiagramLink Method
Diagram.AlignPointToGrid Method

Factory.CreateTableNode Method (Point, Size, Int32, Int32)

Creates a new TableNode instance at the specified position with the specified size and with the specified number of columns and rows, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode CreateTableNode (
    Point location,
    Size size,
    int columns,
    int rows
)
```

Visual Basic Copy Code

```
Public Function CreateTableNode( _
    location As Point, _
    size As Size, _
    columns As Integer, _
    rows As Integer _
) As TableNode
```

Parameters

location
The position of the new table.

size
The size of the new table.

columns
The initial number of columns in the table.

rows
The initial number of rows in the table.

Return Value

The newly created TableNode instance.

Remarks

The method creates a table node at the position specified by *location*. The size of the node is set by *size*. The initial number of columns and rows in the table is specified by *columns* and *rows* respectively. For further customization of the node, use the properties and methods of the TableNode class.

See Also

CreateTableNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateDiagramLink Method
Diagram.AlignPointToGrid Method

Factory.CreateTableNode Method (Rect)

Creates a new TableNode instance with the specified dimensions and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode CreateTableNode (
    Rect bounds
)
```

Visual Basic Copy Code

```
Public Function CreateTableNode( _
    bounds As Rect _
) As TableNode
```

Parameters

bounds
The bounding rectangle of the new table.

Return Value

The newly created TableNode instance.

Remarks

The method creates a table node with dimensions specified by *bounds*. For further customization of the node, use the properties and methods of the TableNode class.

See Also

CreateTableNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateDiagramLink Method
Diagram.AlignPointToGrid Method

Factory.CreateTableNode Method (Rect, Int32, Int32)

Creates a new TableNode instance with the specified dimensions and with the specified number of columns and rows, and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode CreateTableNode (
    Rect bounds,
    int columns,
    int rows
)
```

Visual Basic Copy Code

```
Public Function CreateTableNode( _
    bounds As Rect, _
    columns As Integer, _
    rows As Integer _
) As TableNode
```

Parameters

bounds
The bounding rectangle of the new table.

columns
The initial number of columns in the table.

rows
The initial number of rows in the table.

Return Value

The newly created TableNode instance.

Remarks

The method creates a table node with dimensions specified by *bounds*. The initial number of columns and rows in the table is specified by *columns* and *rows* respectively. For further customization of the node, use the properties and methods of the TableNode class.

See Also

CreateTableNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace
CreateShapeNode Method
CreateDiagramLink Method
Diagram.AlignPointToGrid Method

Factory.CreateTreeViewNode Method

Creates a new TreeViewNode instance.

Overload List

Name	Description
Factory.CreateTreeViewNode (Double, Double, Double, Double)	Creates a new **TreeViewNode** instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.
Factory.CreateTreeViewNode (Point, Size)	Creates a new **TreeViewNode** instance at the specified position with the specified size and adds it to the **Nodes** collection of the underlying diagram.
Factory.CreateTreeViewNode (Rect)	Creates a new **TreeViewNode** instance with the specified dimensions and adds it to the **Nodes** collection of the underlying diagram.

See Also

Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateTreeViewNode Method (Double, Double, Double, Double)

Creates a new TreeViewNode instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewNode CreateTreeViewNode (
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Public Function CreateTreeViewNode( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
) As TreeViewNode
```

Parameters

x
The x-coordinate of the upper-left corner of the new node.

y
The y-coordinate of the upper-left corner of the new node.

width
The width of the new node.

height
The height of the new node.

Return Value

The newly created TreeViewNode instance.

Remarks

The method creates a treeview node at the position specified by *x* and *y*. The size of the node is set by *width* and *height*.

See Also

CreateTreeViewNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateTreeViewNode Method (Point, Size)

Creates a new TreeViewNode instance at the specified position with the specified size and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewNode CreateTreeViewNode (
    Point location,
    Size size
)
```

Visual Basic Copy Code

```
Public Function CreateTreeViewNode( _
    location As Point, _
    size As Size _
) As TreeViewNode
```

Parameters

location

The position of the new node.

size
The size of the new node.

Return Value
The newly created TreeViewNode instance.

Remarks
The method creates a treeview node at the position specified by *location*. The size of the node is set by *size*.

See Also
CreateTreeViewNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

Factory.CreateTreeViewNode Method (Rect)

Creates a new TreeViewNode instance with the specified dimensions and adds it to the Nodes collection of the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewNode CreateTreeViewNode (
    Rect bounds
)
```

Visual Basic Copy Code

```
Public Function CreateTreeViewNode( _
    bounds As Rect _
) As TreeViewNode
```

Parameters
bounds
The bounding rectangle of the new node.

Return Value
The newly created TreeViewNode instance.

Remarks
The method creates a treeview node with dimensions specified by *bounds*.

See Also
CreateTreeViewNode Method Overload List
Factory Members
Factory Class
MindFusion.Diagramming Namespace

FitSize Enumeration

Specifies how the ResizeToFitText method should resize a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

Copy Code

```
public enum FitSize
```

Visual Basic Copy Code

```
Public Enum FitSize
```

Members

	Member name	Description
	KeepWidth	Specifies that the original node width should remain intact.
	KeepHeight	Specifies that the original node height should remain intact.
	KeepRatio	Allows changing both the width and height of a node, but keep the original width/height ratio if possible.

See Also

MindFusion.Diagramming Namespace

FreeFormHandle Class

Represents an adjustment handle used to modify the outline of a FreeFormNode.

Syntax

C# Copy Code

```
public class FreeFormHandle : AdjustmentHandle
```

Visual Basic Copy Code

```
Public Class FreeFormHandle
    Inherits AdjustmentHandle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.AdjustmentHandle
    MindFusion.Diagramming.FreeFormHandle

See Also

FreeFormHandle Members
MindFusion.Diagramming Namespace

FreeFormHandle Members

Represents an adjustment handle used to modify the outline of a FreeFormNode.

The following tables list the members exposed by the FreeFormHandle type.

Public Constructors

	Name	Description
	FreeFormHandle	Initializes a new instance of the **FreeFormHandle** class.

Public Properties

	Name	Description
	PointIndex	Gets or sets the index of a point from FreeFormNode.Points collection that is being modified through this handle.

Public Methods

	Name	Description
	IsMoveHandle	Returns a value indicating whether this handle is used to move items. (Inherited from AdjustmentHandle.)
	IsResizeHandle	Gets whether this adjustment handle resizes the item. (Inherited from AdjustmentHandle.)
	IsRotationHandle	Gets whether this adjustment handle changes item's rotation angle. (Inherited from AdjustmentHandle.)

See Also

FreeFormHandle Class
MindFusion.Diagramming Namespace

FreeFormHandle Constructor

Initializes a new instance of the FreeFormHandle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public FreeFormHandle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

FreeFormHandle Class
MindFusion.Diagramming Namespace

FreeFormHandle Properties

FreeFormHandle.PointIndex Property

Gets or sets the index of a point from FreeFormNode.Points collection that is being modified through this handle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int PointIndex { get; set; }
```

Visual Basic Copy Code

```
Public Property PointIndex As Integer
```

Property Value

An integer value.

See Also

FreeFormHandle Members
FreeFormHandle Class
MindFusion.Diagramming Namespace

FreeFormNode Class

A **FreeFormNode** collects all points from users' mouse or touch input and displays them as node's outline.

Syntax

C# Copy Code

```
public class FreeFormNode : DiagramNode
```

Visual Basic Copy Code

```
Public Class FreeFormNode
    Inherits DiagramNode
```

Remarks

To let users draw free-form nodes interactively, set **Behavior** to DrawFreeForms or **LinkFreeForms**. Use the Points property of **FreeFormNode** to get or set outline points programmatically. If the Closed property is set, the node is drawn as a closed shape and its interior filled, or otherwise the node is drawn as a polyline. If the distance between first and last points drawn by user is shorter than AutoCloseDistance, the node's **Closed** property is automatically set to **true**. **AutoCloseDistance** default value is **float.MaxValue**, so free-form nodes are always closed.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.FreeFormNode

See Also

FreeFormNode Members
MindFusion.Diagramming Namespace

FreeFormNode Members

A FreeFormNode collects all points from users' mouse or touch input and displays them as node's outline.

The following tables list the members exposed by the **FreeFormNode** type.

Public Constructors

	Name	Description
	FreeFormNode	Overloaded.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Closed	Gets or sets if the node's shape is closed.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	ConvertToShape	Gets or sets whether the **FreeFormNode** currently being drawn should be replaced by a matching ShapeNode.
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)

	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Points	Gets or sets the node's outline points.
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed inside this node.
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)

	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded.
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	Draws the node using the specified IGraphics surface.
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramNode.DrawHandles override.
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	Draws the shadow of this node on the specified **IGraphics** surface.

	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline.
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	DiagramNode.GetOutlinePoly override.
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramNode.HitTestHandle override.
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)

	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	StandardShapeReplacement	Returns a replacement **ShapeNode** that most closely matches this node's position and outline shape.
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	UpdateFromPoints	Updates the node's internal state after its points have changed.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramNode.AllowCreate override.
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramNode.CancelModify override.
	CompleteCreate	DiagramNode.CompleteCreate override.
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override.

	CreateState	DiagramNode.CreateState override.
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object.
	LoadFromXml	Loads the item content from an XML element.
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	DiagramNode.OnUpdateBounds override.
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramNode.RestoreProperties override.
	RestoreState	DiagramNode.RestoreState override.

	SaveProperties	DiagramNode.SaveProperties override.
	SaveState	DiagramNode.SaveState override.
	SaveToJson	Saves the node as a JSON object.
	SaveToXml	Saves the item content into an XML element.
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramNode.StartCreate override.
	StartModify	DiagramNode.StartModify override.
	UpdateCreate	DiagramNode.UpdateCreate override.
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override.

See Also

FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode Constructor

Overload List

Name	Description
FreeFormNode ()	Initializes a new instance of the FreeFormNode class.
FreeFormNode (Diagram)	Initializes a new instance of the **FreeFormNode** class.

FreeFormNode (FreeFormNode)	Initializes a new instance of the **FreeFormNode** class using the specified node as a prototype.

See Also

FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode Constructor ()

Initializes a new instance of the FreeFormNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public FreeFormNode ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

FreeFormNode Constructor Overload List
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode Constructor (Diagram)

Initializes a new instance of the FreeFormNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public FreeFormNode (
    Diagram parent
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram _
)
```

Parameters

parent
The Diagram from which to obtain default values for the node properties.

See Also

FreeFormNode Constructor Overload List
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode Constructor (FreeFormNode)

Initializes a new instance of the FreeFormNode class using the specified node as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public FreeFormNode (
    FreeFormNode prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As FreeFormNode _
)
```

Parameters

prototype
The node to use as a prototype. The properties of the new node are initialized from the property values of this node.

See Also

FreeFormNode Constructor Overload List
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode Methods

FreeFormNode.AllowCreate Method

DiagramNode.AllowCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override bool AllowCreate (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Function AllowCreate( _
    current As Point, _
    ist As InteractionState _
) As Boolean
```

Parameters

current
The current mouse pointer position.

ist
An InteractionState object containing more information about the current state of the operation.

Return Value

true to allow the creation of this node; otherwise, **false**.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.CancelModify Method

DiagramNode.CancelModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void CancelModify (
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub CancelModify( _
    ist As InteractionState _
)
```

Parameters

ist
An InteractionState object containing more information about the current state of the operation.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.CompleteCreate Method

DiagramNode.CompleteCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void CompleteCreate (
    Point end
)
```

Visual Basic Copy Code

```
Protected Overrides Sub CompleteCreate( _
    end As Point _
)
```

Parameters

end
The mouse pointer position.

See Also

FreeFormNode.ContainsPoint Method

Overload List

Name	Description
FreeFormNode.ContainsPoint (Point)	DiagramItem.ContainsPoint override.
FreeFormNode.ContainsPoint (Point, Double)	DiagramItem.ContainsPoint override.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.ContainsPoint Method (Point)

DiagramItem.ContainsPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool ContainsPoint (
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function ContainsPoint( _
    point As Point _
) As Boolean
```

Parameters

point

Return Value

See Also

ContainsPoint Method Overload List
FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.ContainsPoint Method (Point, Double)

DiagramItem.ContainsPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool ContainsPoint (
    Point point,
    double threshold
)
```

Visual Basic Copy Code

```
Public Overrides Function ContainsPoint( _
    point As Point, _
    threshold As Double _
) As Boolean
```

Parameters

point

threshold

Return Value

See Also

ContainsPoint Method Overload List
FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.CreateProperties Method

DiagramItem.CreateProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemProperties CreateProperties ()
```

Visual Basic Copy Code

```
Protected Overrides Function CreateProperties () As DiagramItemProperties
```

Return Value

A DiagramItemProperties-derived object containing a snapshot of the item's properties.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.CreateState Method

DiagramNode.CreateState override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemState CreateState ()
```

Visual Basic

```
Protected Overrides Function CreateState () As DiagramItemState
```

Return Value

An instance of the DiagramItemState class.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.Draw Method

Draws the node using the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override void Draw (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic

```
Public Overrides Sub Draw( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics

An IGraphics object where the node should be rendered.

options

Provides options for the drawing operation.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.DrawHandles Method

DiagramNode.DrawHandles override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override void DrawHandles (
    IGraphics graphics,
    HandlesVisualStyle hstyle,
    RenderOptions options
)
```

Visual Basic

```
Public Overrides Sub DrawHandles( _
    graphics As IGraphics, _
    hstyle As HandlesVisualStyle, _
    options As RenderOptions _
)
```

Parameters

graphics
The IGraphics surface to draw the handles on.

hstyle
The style that should be used to render the handles.

options
Provides options for the drawing operation.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.DrawShadow Method

Draws the shadow of this node on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override void DrawShadow (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic

```
Public Overrides Sub DrawShadow( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
An IGraphics object where the shadow should be rendered.

options
Provides options for the drawing operation.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.GetIntersection Method

Calculates the intersection point between the segment defined by the specified point pair and the node's outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Point GetIntersection (
    Point segmentStart,
    Point segmentEnd
)
```

Visual Basic Copy Code

```
Public Overrides Function GetIntersection( _
    segmentStart As Point, _
    segmentEnd As Point _
) As Point
```

Parameters

segmentStart
The first point of the segment.

segmentEnd
The second point of the segment.

Return Value

A **Microsoft.Maui.Graphics.Point** representing the intersection point.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.GetOutlinePoly Method

DiagramNode.GetOutlinePoly override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override PointCollection GetOutlinePoly (
    bool rendering
)
```

Visual Basic Copy Code

```
Public Overrides Function GetOutlinePoly( _
    rendering As Boolean _
) As PointCollection
```

Parameters

rendering

Return Value

A PointCollection containing the node's outline points.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.HitTestHandle Method

DiagramNode.HitTestHandle override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override AdjustmentHandle HitTestHandle (
    Point point,
    double distance
)
```

Visual Basic Copy Code

```
Public Overrides Function HitTestHandle( _
    point As Point, _
    distance As Double _
) As AdjustmentHandle
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying where to look for a selection handle.

distance
Maximum distance from a handle at which it is still considered as hit.

Return Value

Returns an AdjustmentHandle instance representing the adjustment handle found at the specified location, or **null** if there is none.

Remarks

FreeFormNode can return either a FreeFormHandle to modify the node's outline, or a NodeAdjustmentHandle to move or resize the node.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.LoadFromJson Method

Loads the node from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the node's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.LoadFromXml Method

Loads the item content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** containing the item's data.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.OnUpdateBounds Method

DiagramNode.OnUpdateBounds override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnUpdateBounds (
    bool sizeChanged
)
```

Visual Basic Copy Code

```
Protected Overrides Sub OnUpdateBounds( _
    sizeChanged As Boolean _
)
```

Parameters

sizeChanged

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.RestoreProperties Method

DiagramNode.RestoreProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties-derived object containing a snapshot of the item's properties.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.RestoreState Method

DiagramNode.RestoreState override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreState (
    DiagramItemState state
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreState( _
    state As DiagramItemState _
)
```

Parameters

state

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.SaveProperties Method

DiagramNode.SaveProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties-derived object where the item's properties should be saved.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.SaveState Method

DiagramNode.SaveState override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveState (
    DiagramItemState state
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveState( _
    state As DiagramItemState _
)
```

Parameters

state

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.SaveToJson Method

Saves the node as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject instance to contain the serialized data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.SaveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An XmlElement the item's data should be stored into.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.StandardShapeReplacement Method

Returns a replacement ShapeNode that most closely matches this node's position and outline shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNode StandardShapeReplacement (
    Shape[] shapes,
    Shape defaultShape
)
```

Visual Basic Copy Code

```
Public Function StandardShapeReplacement( _
    shapes() As Shape, _
    defaultShape As Shape _
) As ShapeNode
```

Parameters

shapes

An array of Shapes against which the free-form outline is matched.

defaultShape

The default shape to assign to the node if there is no match.

Return Value

A ShapeNode instance.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.StartCreate Method

DiagramNode.StartCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void StartCreate (
    Point org
)
```

Visual Basic

```
Protected Overrides Sub StartCreate( _
    org As Point _
)
```

Parameters

org
Position of the mouse pointer.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.StartModify Method

DiagramNode.StartModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void StartModify (
    Point org,
    AdjustmentHandle handle,
    InteractionState ist
)
```

Visual Basic

```
Protected Overrides Sub StartModify( _
    org As Point, _
    handle As AdjustmentHandle, _
    ist As InteractionState _
)
```

Parameters

org
Position of the mouse pointer.

handle
The adjustment handle used to modify this node.

ist
An InteractionState object containing more information about the current state of the operation.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.UpdateCreate Method

DiagramNode.UpdateCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void UpdateCreate (
    Point current
)
```

Visual Basic

```
Protected Overrides Sub UpdateCreate( _
    current As Point _
)
```

Parameters

current
Position of the mouse pointer.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.UpdateFromPoints Method

Updates the node's internal state after its points have changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void UpdateFromPoints ()
```

Visual Basic

```
Public Sub UpdateFromPoints ()
```

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.UpdateModify Method

DiagramItem.UpdateModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void UpdateModify (
    Point current,
    InteractionState ist
)
```

Visual Basic

```
Protected Overrides Sub UpdateModify( _
    current As Point, _
```

```
    ist As InteractionState _
)
```

Parameters

current
Position of the mouse pointer.

ist
An InteractionState object containing more information about the current state of the operation.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode Properties

FreeFormNode.Closed Property

Gets or sets if the node's shape is closed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Closed { get; set; }
```

Visual Basic Copy Code

```
Public Property Closed As Boolean
```

Property Value

A boolean value.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.ConvertToShape Property

Gets or sets whether the FreeFormNode currently being drawn should be replaced by a matching ShapeNode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ConvertToShape { get; set; }
```

Visual Basic Copy Code

```
Public Property ConvertToShape As Boolean
```

Property Value

A boolean value.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.Points Property

Gets or sets the node's outline points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<Point> Points { get; set; }
```

Visual Basic Copy Code

```
Public Property Points As List(Of Point)
```

Property Value

A collection of points.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNode.Text Property

Gets or sets the text displayed inside this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Text { get; set; }
```

Visual Basic Copy Code

```
Public Property Text As String
```

Property Value

A string that should be drawn inside this node.

See Also

FreeFormNode Members
FreeFormNode Class
MindFusion.Diagramming Namespace

FreeFormNodeStyle Class

Represents a reusable set of properties that define the appearance of free-form nodes.

Syntax

C# Copy Code

```
public class FreeFormNodeStyle : DiagramNodeStyle
```

Visual Basic Copy Code

```
Public Class FreeFormNodeStyle
    Inherits DiagramNodeStyle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle
      MindFusion.Diagramming.DiagramNodeStyle
        MindFusion.Diagramming.FreeFormNodeStyle

See Also

FreeFormNodeStyle Members
MindFusion.Diagramming Namespace

FreeFormNodeStyle Members

Represents a reusable set of properties that define the appearance of free-form nodes.

The following tables list the members exposed by the FreeFormNodeStyle type.

Public Constructors

	Name	Description
	FreeFormNodeStyle	Initializes a new instance of the **FreeFormNodeStyle** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of the item. (Inherited from DiagramItemStyle.)
	Effects	Gets a list of the effects applied to the node. (Inherited from DiagramNodeStyle.)
	FontFamily	Gets or sets the font family of the item text. (Inherited from DiagramItemStyle.)
	FontSize	Gets or sets the font size of the item text. (Inherited from DiagramItemStyle.)
	FontStyle	Gets or sets the font style of the item text. (Inherited from DiagramItemStyle.)
	FontUnit	Gets or sets the font unit of the item text. (Inherited from DiagramItemStyle.)
	ShadowBrush	Gets or sets the brush used to paint the shadow of the item. (Inherited from DiagramItemStyle.)
	Stroke	Gets or sets the brush used to paint the outline of the item. (Inherited from DiagramItemStyle.)
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items. (Inherited from DiagramItemStyle.)

	StrokeThickness	Gets or sets the thickness of the item's outline, expressed in the current units of measure. (Inherited from DiagramItemStyle.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItemStyle.)

Public Methods

	Name	Description
	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
S	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JsonObject. (Inherited from DiagramNodeStyle.)
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)
	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a JsonObject. (Inherited from DiagramNodeStyle.)
	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

FreeFormNodeStyle Class
MindFusion.Diagramming Namespace

FreeFormNodeStyle Constructor

Initializes a new instance of the FreeFormNodeStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public FreeFormNodeStyle ()
```

Visual Basic — Copy Code

```
Public Sub New ()
```

See Also

FreeFormNodeStyle Class
MindFusion.Diagramming Namespace

FreeFormNodeStyle Events

GlassEffect Class

Represents a glass-like visual effect.

Syntax

C# — Copy Code

```
public sealed class GlassEffect : NodeEffect
```

Visual Basic — Copy Code

```
Public NotInheritable Class GlassEffect
    Inherits NodeEffect
```

Remarks

This effect can be applied to the nodes in a Diagram by adding an instance of this class to the NodeEffects collection.

The following image illustrates the Glass effect applied to an item as well as how the different properties affect the output:



Example

The following example demonstrates how to apply glass effect to the nodes of an existing diagram.

C# — Copy Code

```
diagram.NodeEffects.Add(new GlassEffect());
```

Visual Basic — Copy Code

```
diagram.NodeEffects.Add(New GlassEffect())
```

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.NodeEffect
MindFusion.Diagramming.GlassEffect

See Also

GlassEffect Members
MindFusion.Diagramming Namespace

GlassEffect Members

Represents a glass-like visual effect.

The following tables list the members exposed by the GlassEffect type.

Public Constructors

	Name	Description
	GlassEffect	Initializes a new instance of the **GlassEffect** class.

Public Properties

	Name	Description
	GlowColor	Gets or sets the color of the glow portion of the effect.
	ReflectionColor	Gets or sets the color of the reflection portion of the effect.
	Type	Gets or sets the type of the glass effect.
	UsePenAsGlow	Gets or sets a value indicating whether the glow portion of the effect should be painted with the Pen color of the respective node.

Public Methods

	Name	Description
	Clone	NodeEffect.Clone override.

Public Events

	Name	Description
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation. (Inherited from NodeEffect.)

See Also

GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect Constructor

Initializes a new instance of the GlassEffect class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public GlassEffect ()
```

Visual Basic

```
Public Sub New ()
```

See Also

GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect Methods

GlassEffect.Clone Method

NodeEffect.Clone override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override Object Clone ()
```

Visual Basic

```
Public Overrides Function Clone () As Object
```

Return Value

The copy of this effect.

See Also

GlassEffect Members
GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect Properties

GlassEffect.GlowColor Property

Gets or sets the color of the glow portion of the effect.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Color GlowColor { get; set; }
```

Visual Basic

```
Public Property GlowColor As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** instance. The default is **White**.

Remarks

The recommended value for this property is **Black** (or other dark color) for Type4 glass effects.

See Also

GlassEffect Members
GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect.ReflectionColor Property

Gets or sets the color of the reflection portion of the effect.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Color ReflectionColor { get; set; }
```

Visual Basic Copy Code

```
Public Property ReflectionColor As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** instance. The default is **White**.

See Also

GlassEffect Members
GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect.Type Property

Gets or sets the type of the glass effect.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public GlassEffectType Type { get; set; }
```

Visual Basic Copy Code

```
Public Property Type As GlassEffectType
```

Property Value

A member of the GlassEffectType enumeration. The default is **Type1**.

See Also

GlassEffect Members
GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect.UsePenAsGlow Property

Gets or sets a value indicating whether the glow portion of the effect should be painted with the Pen color of the respective node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool UsePenAsGlow { get; set; }
```

Visual Basic Copy Code

```
Public Property UsePenAsGlow As Boolean
```

Property Value

true to ignore GlowColor and use the color of the pen for the effect glow; otherwise, **false**.

See Also

GlassEffect Members
GlassEffect Class
MindFusion.Diagramming Namespace

GlassEffect Events

GlassEffectType Enumeration

Specifies the type of a GlassEffect node effect.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum GlassEffectType
```

Visual Basic Copy Code

```
Public Enum GlassEffectType
```

Members

	Member name	Description
	Type1	Indicates a glass effect with soft radial glow at the bottom and a sharp reflection at the top.
	Type2	Indicates a glass effect with linear glow at the bottom and sharp reflection at the top.
	Type3	Indicates a glass effect with radial glow at the bottom and smooth reflection and emphasized outline at the top.
	Type4	Indicates a glass effect with sharp reflection at the top and soft glow at the left and right sides.

See Also

MindFusion.Diagramming Namespace

Granularity Enumeration

Specifies the granularity of the obstacles map generated by QuickRouter.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum Granularity
```

Visual Basic Copy Code

```
Public Enum Granularity
```

Members

	Member name	Description
	CoarseGrained	Merge map tiles if possible to create a coarse-grained map. The fewer tiles in the map lead to faster path-finding.
	Normal	Use the map tiles found by the initial division algorithm.
	FineGrained	Reserved for future use.

Remarks

Members of this enumeration can be assigned to the Granularity property of QuickRouter.

See Also

MindFusion.Diagramming Namespace

GridRouter Class

Implements a link routing algorithm that uses a grid to represent the free and occupied regions in the diagram.

Syntax

C# Copy Code

```
public class GridRouter : ILinkRouter
```

Visual Basic Copy Code

```
Public Class GridRouter
    Implements ILinkRouter
```

Remarks

To use this class for link routing simply assign its instance to the LinkRouter property of the Diagram class.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.GridRouter

See Also

GridRouter Members
MindFusion.Diagramming Namespace

GridRouter Members

Implements a link routing algorithm that uses a grid to represent the free and occupied regions in the diagram.

The following tables list the members exposed by the GridRouter type.

Public Constructors

	Name	Description
	GridRouter	Initializes a new instance of the **GridRouter** class.

Public Properties

	Name	Description
	Diagram	Gets or sets the diagram this router is associated with.

Public Methods

	Name	Description
	RouteLink	Implements ILinkRouter.RouteLink.
	RouteLinks	Implements ILinkRouter.RouteLinks.

See Also

GridRouter Class
MindFusion.Diagramming Namespace

GridRouter Constructor

Initializes a new instance of the GridRouter class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public GridRouter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

GridRouter Class
MindFusion.Diagramming Namespace

GridRouter Methods

GridRouter.RouteLink Method

Implements ILinkRouter.RouteLink.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RouteLink (
    DiagramLink link,
    ConnectionPoint originConnection,
    ConnectionPoint destinationConnection,
    bool nowCreating
)
```

Visual Basic Copy Code

```
Public Function RouteLink( _
    link As DiagramLink, _
    originConnection As ConnectionPoint, _
    destinationConnection As ConnectionPoint, _
    nowCreating As Boolean _
) As Boolean
```

Parameters

link
The DiagramLink to route.

originConnection
Represents connection to Origin node, potentially different from link's stored connection when modifying first control point.

destinationConnection
Represents connection to Destination node, potentially different from link's stored connection when modifying last control point.

nowCreating
Indicates whether link is currently being created.

Return Value

true if a path has been found successfully, otherwise false.

See Also

GridRouter Members
GridRouter Class
MindFusion.Diagramming Namespace

GridRouter.RouteLinks Method

Implements ILinkRouter.RouteLinks.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection RouteLinks (
    DiagramLinkCollection links
)
```

Visual Basic Copy Code

```
Public Function RouteLinks( _
    links As DiagramLinkCollection _
) As DiagramLinkCollection
```

Parameters

links
A list of DiagramLink objects that should be routed.

Return Value

A list of links that were not routed successfully.

See Also

GridRouter Members
GridRouter Class
MindFusion.Diagramming Namespace

GridRouter Properties

GridRouter.Diagram Property

Gets or sets the diagram this router is associated with.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Diagram Diagram { get; set; }
```

Visual Basic Copy Code

```
Public Property Diagram As Diagram
```

Property Value

See Also

GridRouter Members
GridRouter Class
MindFusion.Diagramming Namespace

GridStyle Enumeration

Defines values that specify the visual style of the alignment grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum GridStyle
```

Visual Basic Copy Code

```
Public Enum GridStyle
```

Members

	Member name	Description
	Points	The grid is painted as a matrix of points.
	Lines	The grid is represented as a series of crossing horizontal and vertical lines.

See Also

Group Class

Represents a group of items in the diagram document.

Syntax

C# Copy Code

```
public class Group
```

Visual Basic Copy Code

```
Public Class Group
```

Remarks

An hierarchy of items can be built, so that moving an item that is higher in the hierarchy moves all subordinated items too. The immediate subordinates of an item are represented by a **Group** instance. Every group should have a single master item, to which all group members are attached. The master item can be accessed from the MainItem property. The group of items attached to an item is accessed through the SubordinateGroup property of the latter. MasterGroup gets the group in which an item is subordinate.

To create a group, instantiate a **Group** object and add it to the diagram's Groups collection, or call the CreateGroup method passing the master item as argument. A group is automatically destroyed when its master item is deleted.

To destroy a group programmatically, call the Remove method of the **Groups** collection. If AutoDeleteItems is enabled, subordinated items in the group are deleted too. If **AutoDeleteItems** is disabled, subordinated items are just detached from their master and remain intact.

Nodes can be attached to a group in several ways, depending on the type of the master item. If the master is a node, other nodes can be attached by calling the AttachProportional or AttachToCorner methods. If the master is a link, nodes can be attached to it using the AttachToLinkSegment or AttachToLinkPoint methods. The DiagramNode class provides a shortcut AttachTo method, which creates a group if necessary and calls the group's appropriate **AttachTo*** method. Items can be detached from a group by calling the Detach method.

The Expandable property specifies whether expanding or collapsing a tree should propagate from the master node through the subordinated nodes. FollowMasterRotation can be used when the master node is a shape node. If it is enabled, the positions of the subordinated nodes are updated to reflect the RotationAngle of the shape node while it is rotated.

The Tag property allows attaching custom data to any group. The FindGroup methods looks for a groups having a specific tag.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Group

See Also

Group Members
MindFusion.Diagramming Namespace

Group Members

Represents a group of items in the diagram document.

The following tables list the members exposed by the Group type.

Public Constructors

	Name	Description
	Group	Overloaded. Initializes a new instance of the **Group** class.

Public Properties

	Name	Description
	AttachedNodes	Gets a collection of the nodes attached to this group.
	AutoDeleteItems	Gets or sets a value indicating whether the subordinated items in a group are automatically deleted if the group is destroyed.
	Expandable	Gets or sets a value indicating whether expanding/collapsing a tree branch propagates through this group.
	FollowMasterContainment	Gets or sets a value indicating whether the subordinated items should be considered by a ContainerNode when the master item is added to the container.
	FollowMasterRotation	Gets or sets a value indicating whether attached nodes position should follow the RotationAngle of a master node.
	MainItem	Gets or sets the main item of this group.
	Tag	Gets or sets a custom data associated with the group.
	Visible	Gets or sets a value indicating whether the group is visible.

Public Methods

	Name	Description
	AttachAlongLinkLength	Attaches a node to a link and keeps the initial distance between the node and the link origin when the latter is modified. The distance is measured along the link length, regardless of the number and shape of its segments.
	AttachCustom	Attaches a new node to the group.
	AttachProportional	Attaches a new node to the group, in such a way that its area is always occupying certain percentage of the main item's area.
	AttachToCorner	Attaches a node to the specified corner of the main node.
	AttachToEdges	Anchors a node to the specified edges of the master node.
	AttachToFractionalPosition	Attaches specified node to this group, keeping original size intact and maintaining original distance to point specified as fraction of main node's size.
	AttachToLinkPoint	Attaches a node to the specified control point of the link that is **MainItem**.
	AttachToLinkSegment	Attaches a node to the specified segment of the link that is **MainItem**.
	AttachToLongestHSegment	Adds an item to the group, maintaining its relative position to the longest horizontal segment of the master link.
	AttachToSideMiddle	Adds an item to the group, attaching it to the middle point of a side of the master node.
	Detach	Detaches the specified item from this group.
	GetRepaintRect	Gets a rectangle containing the repaint rectangle of the group's master and all attached objects.

See Also

Group Class
MindFusion.Diagramming Namespace

Group Constructor

Initializes a new instance of the Group class.

Overload List

Name	Description
Group (Diagram)	Initializes a new instance of the **Group** class with the specified Diagram as parent.
Group (Group)	Initializes a new instance of the **Group** class using the specified group as a prototype.

See Also

Group Class
MindFusion.Diagramming Namespace

Group Constructor (Diagram)

Initializes a new instance of the Group class with the specified Diagram as parent.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Group (
    Diagram parent
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram _
)
```

Parameters

parent
The Diagram the new group will belong to.

See Also

Group Constructor Overload List
Group Class
MindFusion.Diagramming Namespace

Group Constructor (Group)

Initializes a new instance of the Group class using the specified group as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Group (
    Group prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As Group _
)
```

Parameters

prototype
The group to use as a prototype.

Remarks

This constructor supports the MvcDiagram infrastructure and is not intended to be used directly from your code. Instead use the other constructor or the CreateGroup method of the Factory class.

See Also

Group Constructor Overload List
Group Class
MindFusion.Diagramming Namespace

Group Methods

Group.AttachAlongLinkLength Method

Attaches a node to a link and keeps the initial distance between the node and the link origin when the latter is modified. The distance is measured along the link length, regardless of the number and shape of its segments.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AttachAlongLinkLength (
    DiagramNode node,
    bool relativeDistance
)
```

Visual Basic Copy Code

```
Public Function AttachAlongLinkLength( _
    node As DiagramNode, _
    relativeDistance As Boolean _
) As Boolean
```

Parameters

node
The DiagramNode to be attached.

relativeDistance
true to preserve the absolute distance, false to preserve relative distance.

Return Value

false if the node could not be attached; otherwise, true.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace

Group.AttachCustom Method

Attaches a new node to the group.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AttachCustom (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Function AttachCustom( _
    node As DiagramNode _
) As Boolean
```

Parameters

node
A subordinate node whose position depends on the position of the group master item.

Return Value

true if the node has been attached to this group successfully.

Remarks

This method is used in conjunction with an UpdateGroup override defined in derived item types. The **UpdateGroup** override must set the position of attached nodes by calling their SetBounds method.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace

Group.AttachProportional Method

Attaches a new node to the group, in such a way that its area is always occupying certain percentage of the main item's area.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AttachProportional (
    DiagramNode node,
    double left,
    double top,
    double right,
    double bottom
)
```

Visual Basic Copy Code

```
Public Function AttachProportional( _
    node As DiagramNode, _
    left As Double, _
    top As Double, _
    right As Double, _
    bottom As Double _
) As Boolean
```

Parameters

node
The DiagramNode to be attached.

left
The left side of the node's bounding rectangle, expressed as percentage of the main node width.

top

The top side of the node's bounding rectangle, expressed as percentage of the main node height.

right
The right side of the node's bounding rectangle, expressed as percentage of the main node width.

bottom
The bottom side of the node's bounding rectangle, expressed as percentage of the main node height.

Return Value

false if the node could not be attached; otherwise, **true**.

Remarks

You can use this method when the main item of the group is a node. Use the left, top, right and bottom arguments to specify the position and size of the attached node relative to that of the main item. When moving or resizing the main node, the specified ratio between it and the attached node is preserved.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace
AttachToCorner Method

Group.AttachToCorner Method

Attaches a node to the specified corner of the main node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AttachToCorner (
    DiagramNode node,
    int corner
)
```

Visual Basic Copy Code

```
Public Function AttachToCorner( _
    node As DiagramNode, _
    corner As Integer _
) As Boolean
```

Parameters

node
The DiagramNode to be attached.

corner
The zero-based index of the main node's corner.

Return Value

false if the node could not be attached; otherwise, **true**.

Remarks

You can use this method if the main item of the group is a node. When the specified corner position of the main node's bounding rectangle is changed, the position of the attached node is also changed and the distance between them is preserved.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace
AttachProportional Method

Group.AttachToEdges Method

Anchors a node to the specified edges of the master node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AttachToEdges (
    DiagramNode node,
    GroupAnchorStyles anchorStyles
)
```

Visual Basic Copy Code

```
Public Function AttachToEdges( _
    node As DiagramNode, _
    anchorStyles As GroupAnchorStyles _
) As Boolean
```

Parameters

node
The DiagramNode to be attached.

anchorStyles
A member of the GroupAnchorStyles enumeration.

Return Value

false if the node could not be attached; otherwise, **true**.

Remarks

This method can be used only if the master item in the group is a node. Anchoring works in a manner similar to how child controls in a form can be anchored to the form edges. When the master node position changes, the position of the attached node is also changed so that the initial distance between the specified edges is preserved.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace
AttachProportional Method

Group.AttachToFractionalPosition Method

Attaches specified node to this group, keeping original size intact and maintaining original distance to point specified as fraction of main node's size.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AttachToFractionalPosition (
    DiagramNode node,
    double xPercent,
    double yPercent
)
```

Visual Basic Copy Code

```
Public Function AttachToFractionalPosition( _
    node As DiagramNode, _
    xPercent As Double, _
```

```
    yPercent As Double _
) As Boolean
```

Parameters

node
The DiagramNode to be attached.

xPercent
Percentage of main node's width.

yPercent
Percentage of main node's height.

Return Value

true if specified node was attached to the group, or **false** otherwise.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace

Group.AttachToLinkPoint Method

Attaches a node to the specified control point of the link that is MainItem.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AttachToLinkPoint (
    DiagramNode node,
    int point
)
```

Visual Basic Copy Code

```
Public Function AttachToLinkPoint( _
    node As DiagramNode, _
    point As Integer _
) As Boolean
```

Parameters

node
The DiagramNode to be attached.

point
The link point to which the node should be attached.

Return Value

false if the node could not be attached; otherwise, **true**.

Remarks

You can use this method when the main item of the group is a link. When the user moves the link control point whose index is passed as argument, the attached node is also moved with the same translation vector as that of the control point.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace
AttachToLinkSegment Method

Group.AttachToLinkSegment Method

Attaches a node to the specified segment of the link that is MainItem.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AttachToLinkSegment (
    DiagramNode node,
    int segment
)
```

Visual Basic Copy Code

```
Public Function AttachToLinkSegment( _
    node As DiagramNode, _
    segment As Integer _
) As Boolean
```

Parameters

node
The DiagramNode to be attached.

segment
The link segment to which the node will be attached.

Return Value

false if the node could not be attached; otherwise, **true**.

Remarks

Call this method when the main item of the group is a link. When the specified link segment changes its position, the position of the attached node is also changed, so their mutual distance and orientation are preserved.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace
AttachToLinkPoint Method

Group.AttachToLongestHSegment Method

Adds an item to the group, maintaining its relative position to the longest horizontal segment of the master link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AttachToLongestHSegment (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Function AttachToLongestHSegment( _
    node As DiagramNode _
) As Boolean
```

Parameters

node

The object to attach to the current group.

Return Value

false if the node could not be attached; otherwise, **true**.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace

Group.AttachToSideMiddle Method

Adds an item to the group, attaching it to the middle point of a side of the master node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AttachToSideMiddle (
    DiagramNode node,
    int side
)
```

Visual Basic Copy Code

```
Public Function AttachToSideMiddle( _
    node As DiagramNode, _
    side As Integer _
) As Boolean
```

Parameters

node
The object to attach to the current group.

side
The corner to attach to.

Return Value

false if the node could not be attached; otherwise, **true**.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace

Group.Detach Method

Detaches the specified item from this group.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Detach (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Function Detach( _
    item As DiagramItem _
) As Boolean
```

Parameters

item
The DiagramItem to be detached.

Return Value

false if the item could not be detached; otherwise, **true**.

Remarks

Call this method to remove a item from this group. Note that this method does not delete the item, but just detaches it.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace
AttachToLinkSegment Method
AttachToLinkPoint Method
AttachProportional Method
AttachToCorner Method

Group.GetRepaintRect Method

Gets a rectangle containing the repaint rectangle of the group's master and all attached objects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect GetRepaintRect ()
```

Visual Basic Copy Code

```
Public Function GetRepaintRect () As Rect
```

Return Value

A **Microsoft.Maui.Graphics.Rect** that represents the smallest rectangle containing all diagram items related directly or indirectly to this group.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace

Group Properties

Group.AttachedNodes Property

Gets a collection of the nodes attached to this group.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ReadOnlyDiagramNodeCollection AttachedNodes { get; }
```

Visual Basic — Copy Code

```
Public ReadOnly Property AttachedNodes As ReadOnlyDiagramNodeCollection
```

Property Value

A collection that references all nodes attached to the group.

Remarks

The MainItem is not included in this collection.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace
MainItem Property

Group.AutoDeleteItems Property

Gets or sets a value indicating whether the subordinated items in a group are automatically deleted if the group is destroyed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# — Copy Code

```
public bool AutoDeleteItems { get; set; }
```

Visual Basic — Copy Code

```
Public Property AutoDeleteItems As Boolean
```

Property Value

true to delete subordinated items automatically; otherwise, **false**. The default is **false**.

Remarks

A group is destroyed explicitly by calling the Remove method of the Groups collection. A group is destroyed implicitly when its MainItem is deleted.

If a group is destroyed and this property is disabled, all subordinated items remain intact in the diagram, however they are no longer grouped. If a group is destroyed and this property is enabled, all subordinated items are deleted.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace

Group.Expandable Property

Gets or sets a value indicating whether expanding/collapsing a tree branch propagates through this group.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# — Copy Code

```
public bool Expandable { get; set; }
```

Visual Basic

```
Public Property Expandable As Boolean
```

Property Value

true if expanding or collapsing of a tree branch should propagate through the group members; otherwise, **false**. The default is **false**.

Remarks

By default expanding or collapsing tree branches is done by following the links between the branch nodes. If there are not links going out from a node, the expanding or collapsing stops there. However, if a node has a group of subordinated nodes, it could be desirable to continue expanding through them. To enable that, set the group's **Expandable** property to **true**.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace
Diagram.RecursiveExpand Property
Diagram.NodesExpandable Property

Group.FollowMasterContainment Property

Gets or sets a value indicating whether the subordinated items should be considered by a ContainerNode when the master item is added to the container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool FollowMasterContainment { get; set; }
```

Visual Basic

```
Public Property FollowMasterContainment As Boolean
```

Property Value

true if the whole group is contained in the ContainerNode; otherwise, **false**. The default is **false**.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace

Group.FollowMasterRotation Property

Gets or sets a value indicating whether attached nodes position should follow the RotationAngle of a master node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public GroupRotation FollowMasterRotation { get; set; }
```

Visual Basic

```
Public Property FollowMasterRotation As GroupRotation
```

Property Value

A member of the GroupRotation enumeration. The default is **IgnoreRotation**.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace

Group.MainItem Property

Gets or sets the main item of this group.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem MainItem { get; set; }
```

Visual Basic Copy Code

```
Public Property MainItem As DiagramItem
```

Property Value

An instance of the DiagramItem class that represents the main item.

Remarks

Each group has an item which the subordinated items follow when it is moved. This main item can be a node or a link. The main item is passed as argument to the CreateGroup method.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace
AttachedNodes Property

Group.Tag Property

Gets or sets a custom data associated with the group.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Tag { get; set; }
```

Visual Basic Copy Code

```
Public Property Tag As Object
```

Property Value

An instance of a **System.Object**-derived class that represents the tag.

Remarks

This property holds data that you define. MvcDiagram does not use it in any way. Possible usage might be to store an integer identifier or a reference to a larger structure.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace
Diagram.FindGroup Method
Diagram.SerializeTag Event

Group.Visible Property

Gets or sets a value indicating whether the group is visible.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Visible { get; set; }
```

Visual Basic Copy Code

```
Public Property Visible As Boolean
```

Property Value

true if the group is visible; otherwise, **false**.

Remarks

Assigning a value to this property sets the Visible property of all items in the group.

See Also

Group Members
Group Class
MindFusion.Diagramming Namespace

Group.Attachment Class

Encapsulates attachment information.

Syntax

C# Copy Code

```
public class Group.Attachment
```

Visual Basic Copy Code

```
Public Class Group.Attachment
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Group.Attachment

See Also

Group.Attachment Members
MindFusion.Diagramming Namespace

Group.Attachment Members

Encapsulates attachment information.

The following tables list the members exposed by the Group.Attachment type.

Public Constructors

	Name	Description
	Group.Attachment	Initializes a new instance of the Attachment class.

Public Properties

	Name	Description
	Data	Gets or sets the attachment data.
	Node	Gets or sets the attached node.
	Percents	Gets a sets percent information for proportional attachments.
	Traits	Contains properties specific to each layout algorithm.
	Type	Gets or sets the attachment type.

Public Methods

	Name	Description
	Clone	Creates an exact copy of this attachment.

See Also

Group.Attachment Class
MindFusion.Diagramming Namespace

Group.Attachment Constructor

Initializes a new instance of the Attachment class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Group.Attachment ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

Group.Attachment Class
MindFusion.Diagramming Namespace

Group.Attachment Methods

Group.Attachment.Clone Method

Creates an exact copy of this attachment.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Group.Attachment Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As Group.Attachment
```

Return Value

See Also

Group.Attachment Members
Group.Attachment Class
MindFusion.Diagramming Namespace

Group.Attachment Properties

Group.Attachment.Data Property

Gets or sets the attachment data.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Data { get; set; }
```

Visual Basic Copy Code

```
Public Property Data As Integer
```

Property Value

Remarks

The interpretation of this value depends on the type of the attachment.

See Also

Group.Attachment Members
Group.Attachment Class
MindFusion.Diagramming Namespace

Group.Attachment.Node Property

Gets or sets the attached node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Node { get; set; }
```

Visual Basic Copy Code

```
Public Property Node As DiagramNode
```

Property Value

See Also

Group.Attachment Members
Group.Attachment Class
MindFusion.Diagramming Namespace

Group.Attachment.Percents Property

Gets a sets percent information for proportional attachments.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect Percents { get; set; }
```

Visual Basic Copy Code

```
Public Property Percents As Rect
```

Property Value

See Also

Group.Attachment Members
Group.Attachment Class
MindFusion.Diagramming Namespace

Group.Attachment.Traits Property

Contains properties specific to each layout algorithm.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IDictionary Traits { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Traits As IDictionary
```

Property Value

See Also

Group.Attachment.Type Property

Gets or sets the attachment type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AttachTo Type { get; set; }
```

Visual Basic Copy Code

```
Public Property Type As AttachTo
```

Property Value

See Also

Group.Attachment Members
Group.Attachment Class
MindFusion.Diagramming Namespace

GroupAnchorStyles Enumeration

Specifies the master node edges to which a subordinate node can be anchored.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum GroupAnchorStyles
```

Visual Basic Copy Code

```
Public Enum GroupAnchorStyles
```

Members

	Member name	Description
	Left	Specifies the left edge.
	Top	Specifies the top edge.
	Right	Specifies the right edge.
	Bottom	Specifies the bottom edge.

See Also

MindFusion.Diagramming Namespace

GroupCollection Class

Represents a strongly typed collection of Group objects.

Syntax

C# Copy Code

```
public class GroupCollection : CollectionBase<Group>
```

Visual Basic Copy Code

```
Public Class GroupCollection
    Inherits CollectionBase(Of Group)
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.CollectionBase(Group)
    MindFusion.Diagramming.GroupCollection

See Also

GroupCollection Members
MindFusion.Diagramming Namespace

GroupCollection Members

Represents a strongly typed collection of Group objects.

The following tables list the members exposed by the GroupCollection type.

Public Constructors

	Name	Description
	GroupCollection	Initializes a new instance of the **GroupCollection** class.

Protected Fields

	Name	Description
	groupEventArgs	A ItemEventArgs instance used for passing arguments to the event handlers for the events of this collection. Used to avoid creating many small objects that would burden the garbage collector.
	InnerList	Gets a **List<T>** containing the list of elements in the collection instance. (Inherited from CollectionBase.)
	notifyParent	If enabled, the collection raises events to let their containing objects handle the addition or removal of items.

Public Properties

	Name	Description
	Count	Gets the number of elements actually contained in the collection. (Inherited from CollectionBase.)
	IsReadOnly	Gets a value indicating whether the collection is read-only. (Inherited from CollectionBase.)
	Item	Gets or sets the element at the specified index. (Inherited from CollectionBase.)

Public Methods

	Name	Description
	Add	Adds an object to the end of the collection. (Inherited from CollectionBase.)
	Add	Adds the specified group to the collection.
	Clear	Removes all groups from the collection.
	Clone	Creates a shallow copy of this collection.
	Contains	Determines whether an element is in the collection. (Inherited from CollectionBase.)
	CopyTo	Copies the entire collection to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from CollectionBase.)
	GetAt	The zero-based index of the group to retrieve.
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from CollectionBase.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire collection. (Inherited from CollectionBase.)
	Insert	Inserts an element into the collection at the specified index. (Inherited from CollectionBase.)
	Insert	Inserts the specified group in the collection at the specified position.
	Remove	Removes the first occurrence of a specific object from the collection. (Inherited from CollectionBase.)
	Remove	Removes the specified group from the collection.
	RemoveAt	Removes the diagram item from the collection.
	Sort	Overloaded. (Inherited from CollectionBase.)

Protected Methods

	Name	Description
	RaiseAdding	Raises the Adding event.
	RaiseInserting	Raises the Inserting event.
	RaiseRemoving	Raises the Removing event.

See Also

GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection Fields

GroupCollection.groupEventArgs Field

A ItemEventArgs instance used for passing arguments to the event handlers for the events of this collection. Used to avoid creating many small objects that would burden the garbage collector.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected GroupEventArgs groupEventArgs
```

Visual Basic

```
Protected groupEventArgs As GroupEventArgs
```

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection.notifyParent Field

If enabled, the collection raises events to let their containing objects handle the addition or removal of items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected bool notifyParent
```

Visual Basic

```
Protected notifyParent As Boolean
```

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection Constructor

Initializes a new instance of the GroupCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public GroupCollection ()
```

Visual Basic

```
Public Sub New ()
```

See Also

GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection Methods

GroupCollection.Add Method

Adds the specified group to the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Add (
    Group group
)
```

Visual Basic Copy Code

```
Public Overrides Sub Add( _
    group As Group _
)
```

Parameters

group
The Group object to add to the collection.

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection.Clear Method

Removes all groups from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Clear ()
```

Visual Basic Copy Code

```
Public Overrides Sub Clear ()
```

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection.Clone Method

Creates a shallow copy of this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public GroupCollection Clone ()
```

Visual Basic — Copy Code

```
Public Function Clone () As GroupCollection
```

Return Value

An instance of the GroupCollection class.

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection.GetAt Method

The zero-based index of the group to retrieve.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# — Copy Code

```
public Group GetAt (
    int index
)
```

Visual Basic — Copy Code

```
Public Function GetAt( _
    index As Integer _
) As Group
```

Parameters

index
The zero-based index of the group to retrieve.

Return Value

An instance of the Group class that represents the node at the specified position.

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection.Insert Method

Inserts the specified group in the collection at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# — Copy Code

```
public override void Insert (
    int index,
    Group group
)
```

Visual Basic

```
Public Overrides Sub Insert( _
    index As Integer, _
    group As Group _
)
```

Parameters

index
The zero-based index at which to insert the group.

group
The Group object to add to the collection.

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection.RaiseAdding Method

Raises the Adding event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected void RaiseAdding (
    Group group
)
```

Visual Basic

```
Protected Sub RaiseAdding( _
    group As Group _
)
```

Parameters

group
The group that is being added.

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection.RaiseInserting Method

Raises the Inserting event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected void RaiseInserting (
    Group group,
    int index
)
```

Visual Basic

```
Protected Sub RaiseInserting( _
    group As Group, _
    index As Integer _
)
```

Parameters

group
The group that is being added.

index
Index of the new group.

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection.RaiseRemoving Method

Raises the Removing event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected void RaiseRemoving (
    Group group
)
```

Visual Basic

```
Protected Sub RaiseRemoving( _
    group As Group _
)
```

Parameters

group
The group that is being removed.

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection.Remove Method

Removes the specified group from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override bool Remove (
    Group group
)
```

Visual Basic

```
Public Overrides Function Remove( _
    group As Group _
) As Boolean
```

Parameters

group
The Group object to remove from the collection.

Return Value

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupCollection.RemoveAt Method

Removes the diagram item from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override void RemoveAt (
    int index
)
```

Visual Basic

```
Public Overrides Sub RemoveAt( _
    index As Integer _
)
```

Parameters

index
Index of the object that should be removed from this collection.

See Also

GroupCollection Members
GroupCollection Class
MindFusion.Diagramming Namespace

GroupEventArgs Class

Contains the arguments passed to handlers of group-related events, such as GroupDestroyed.

Syntax

C#

```
public class GroupEventArgs : EventArgs
```

Visual Basic

```
Public Class GroupEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.GroupEventArgs

See Also

GroupEventArgs Members
MindFusion.Diagramming Namespace

GroupEventArgs Members

Contains the arguments passed to handlers of group-related events, such as GroupDestroyed.

The following tables list the members exposed by the GroupEventArgs type.

Public Constructors

	Name	Description
	GroupEventArgs	Initializes a new instance of the **GroupEventArgs** class.

Public Properties

	Name	Description
	Group	Gets the group for which the event was raised.
	Index	Gets an integer index related to the event.

See Also

GroupEventArgs Class
MindFusion.Diagramming Namespace

GroupEventArgs Constructor

Initializes a new instance of the GroupEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public GroupEventArgs (
    Group group
)
```

Visual Basic Copy Code

```
Public New ( _
    group As Group _
)
```

Parameters

group
The Group related to the event.

See Also

GroupEventArgs Properties

GroupEventArgs.Group Property

Gets the group for which the event was raised.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Group Group { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Group As Group
```

Property Value

An instance of the Group class.

See Also

GroupEventArgs Members
GroupEventArgs Class
MindFusion.Diagramming Namespace

GroupEventArgs.Index Property

Gets an integer index related to the event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Index { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Index As Integer
```

Property Value

An integer value.

See Also

GroupEventArgs Members
GroupEventArgs Class
MindFusion.Diagramming Namespace

GroupRotation Enumeration

Defines values that indicate how rotation of a group's master node affects the nodes attached to that group.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum GroupRotation
```

Visual Basic Copy Code

```
Public Enum GroupRotation
```

Members

	Member name	Description
	IgnoreRotation	Attached nodes ignore the rotation angle applied to the master node.
	FerrisWheel	Attached nodes are moved to a position corresponding to the master node's rotation angle, but they are not rotated.
	RotateChildren	Attached nodes are rotated by the RotationAngle of the group's master node and around its center.

See Also

MindFusion.Diagramming Namespace

HandlesStyle Enumeration

Enumerates possible visual styles for frames that are drawn around selected nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum HandlesStyle
```

Visual Basic Copy Code

```
Public Enum HandlesStyle
```

Members

	Member name	Description
	Invisible	There is no visible difference between selected and unselected node.
	SquareHandles	If selected, the node is drawn with square modification handles.
	DashFrame	If selected, a node is drawn with a dashed frame. The frame enables resizing the node. The center of the node allows moving it and the edge area enables creating links.
	HatchFrame	If selected, the item is drawn with hatched frame. Modifications can be done as with **DashFrame**.
	HatchHandles	If selected, a node is drawn with both hatched frame and modification handles. The frame enable moving the node, and the handles enable resizing. Links are created from any point of the interior.
	HatchHandles2	Looks like **HatchHandles**, but allows moving the node from the interior area. Links are created from the points near the edge.

	HatchHandles3	Behaves like **HatchHandles2**, but paints the selection frame using a denser brush pattern.
	MoveOnly	If a node is selected, it can be only moved. Best used with **ModificationStart** set to AutoHandles.
	EasyMove	Any point of the interior of a node enables moving the node, except a small area in the center that allows creating links. Best used with **ModificationStart** set to **AutoHandles**.
	SquareHandles2	Square resize handles are drawn around a selected node. There isn't a move handle in the center as with the **SquareHandles** style, but nodes can be moved by dragging their interior or caption areas. Links can be created by drawing from near the edges of a node.
	Custom	MvcDiagram raises the DrawAdjustmentHandles and HitTestAdjustmentHandles events to let your application perform custom drawing and hit testing of selection handles.
	InvisibleMove	There aren't any selection handles displayed. The selected node can be only moved, but not resized.
	RoundAndSquare	Corner resize handles are round, remaining handles are square.
	RoundAndSquare2	Corner resize handles are round, remaining handles are square. In addition, there is a dashed frame drawn around selected nodes.

Remarks

The selection style can be set independently for individual nodes and links through their **DiagramNode.HandlesStyle** and DiagramLink.HandlesStyle properties. For most of these styles, it is recommended to set **ModificationStart** to **AutoHandles**.

See Also

MindFusion.Diagramming Namespace

HandlesVisualStyle Class

Defines the Pen and Brush objects used to draw items' adjustment handles.

Syntax

C# Copy Code

```
public class HandlesVisualStyle
```

Visual Basic Copy Code

```
Public Class HandlesVisualStyle
```

Remarks

The component uses several different **HandlesVisualStyle** objects depending on the context. They can be accessed through the ActiveItemHandlesStyle, SelectedItemHandlesStyle and DisabledHandlesStyle properties of Diagram.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.HandlesVisualStyle

See Also

HandlesVisualStyle Members
MindFusion.Diagramming Namespace

HandlesVisualStyle Members

Defines the Pen and Brush objects used to draw items' adjustment handles.

The following tables list the members exposed by the HandlesVisualStyle type.

Public Constructors

	Name	Description
	HandlesVisualStyle	Initializes a new instance of the HandlesVisualStyle class.

Public Properties

	Name	Description
	ControlPointBrush	Gets or sets a **Brush** used to fill the interior of shape control points.
	DashPen	Gets or sets a **Pen** used to draw dashed selection frames.
	HandleBrush	Gets or sets a **Brush** used to fill the interior of selection handles.
	HandlePen	Gets or sets a **Pen** used to draw the border of selection handles.
	HatchBrush	Gets or sets a **Brush** used to draw hatch frames.
	PatternBrush	Gets or sets a **Brush** used to draw selection frames for the MoveOnly style.
	RotationHandleBrush	Gets or sets a **Brush** used to fill the interior of the rotation handle.

Public Methods

	Name	Description
	LoadFromXml	Loads the style from XML.
	SaveToXml	Saves the style to XML.

Protected Methods

	Name	Description
	LoadFromJson	Loads the style content from a JSON object.
	SaveToJson	Saves the style content into a JsonObject.

See Also

HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle Constructor

Initializes a new instance of the HandlesVisualStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HandlesVisualStyle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle Methods

HandlesVisualStyle.LoadFromJson Method

Loads the style content from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the style's data.

context
A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.LoadFromXml Method

Loads the style from XML.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    string elementName,
    string colorElementName,
    XmlElement parentElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    elementName As String, _
    colorElementName As String, _
    parentElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

elementName

The name of the element that describes the style.

colorElementName

The name of an element that contains a fallback color.

parentElement

The parent XML element.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.SaveToJson Method

Saves the style content into a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject the style's data should be stored into.

context

A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.SaveToXml Method

Saves the style to XML.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToXml (
    string elementName,
    XmlElement parentElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Sub SaveToXml( _
    elementName As String, _
    parentElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

elementName
The name of the element that describes the style.

parentElement
The parent XML element.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle Properties

HandlesVisualStyle.ControlPointBrush Property

Gets or sets a Brush used to fill the interior of shape control points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush ControlPointBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property ControlPointBrush As Brush
```

Property Value

An instance of a Brush -derived class. The default is solid-color yellow brush.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.DashPen Property

Gets or sets a Pen used to draw dashed selection frames.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen DashPen { get; set; }
```

Visual Basic Copy Code

```
Public Property DashPen As Pen
```

Property Value

An instance of the Pen class.

Remarks

This pen is used to draw selection frame around items whose **HandlesStyle** property is set to DashFrame.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.HandleBrush Property

Gets or sets a Brush used to fill the interior of selection handles.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush HandleBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property HandleBrush As Brush
```

Property Value

An instance of a Brush -derived class.

Remarks

This brush is used to fill the interior of adjustment handles rendered for most of the HandlesStyle types.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.HandlePen Property

Gets or sets a Pen used to draw the border of selection handles.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen HandlePen { get; set; }
```

Visual Basic Copy Code

```
Public Property HandlePen As Pen
```

Property Value

An instance of the Pen class.

Remarks

This pen is used to draw the borders of adjustment handles rendered for most of the HandlesStyle types.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.HatchBrush Property

Gets or sets a Brush used to draw hatch frames.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush HatchBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property HatchBrush As Brush
```

Property Value

An instance of a Brush -derived class.

Remarks

This brush is used to draw the hatch pattern for HandlesStyle types that display hatch frames.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.PatternBrush Property

Gets or sets a Brush used to draw selection frames for the MoveOnly style.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush PatternBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property PatternBrush As Brush
```

Property Value

An instance of a Brush -derived class.

Remarks

This brush is used to render the dot pattern for the HatchHandles3 and **MoveOnly HandlesStyle** types.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HandlesVisualStyle.RotationHandleBrush Property

Gets or sets a Brush used to fill the interior of the rotation handle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush RotationHandleBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property RotationHandleBrush As Brush
```

Property Value

A Brush instance. The default value is **null**.

Remarks

If this property is set to **null**, the rotation brush is painted using HandleBrush.

See Also

HandlesVisualStyle Members
HandlesVisualStyle Class
MindFusion.Diagramming Namespace

HitTestEventArgs Class

Contains the arguments passed to the HitTestAdjustmentHandles event handlers.

Syntax

C# Copy Code

```
public class HitTestEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class HitTestEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.HitTestEventArgs

See Also

HitTestEventArgs Members
MindFusion.Diagramming Namespace

HitTestEventArgs Members

Contains the arguments passed to the HitTestAdjustmentHandles event handlers.

The following tables list the members exposed by the HitTestEventArgs type.

Public Constructors

	Name	Description
	HitTestEventArgs	Initializes a new instance of the **HitTestEventArgs** class.

Public Properties

	Name	Description
	HitResult	Gets or sets the adjustment handle that has been hit.
	Item	Gets the item whose adjustment handles are being hit-tested.
	MousePosition	Gets the current cursor position, expressed in document coordinates.

See Also

HitTestEventArgs Class
MindFusion.Diagramming Namespace

HitTestEventArgs Constructor

Initializes a new instance of the HitTestEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HitTestEventArgs (
    DiagramItem item,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public New ( _
    item As DiagramItem, _
    mousePosition As Point _
)
```

Parameters

item

The DiagramItem related to the event.

mousePosition

The cursor location at the time the event is raised, expressed in document coordinates.

See Also

HitTestEventArgs Class
MindFusion.Diagramming Namespace

HitTestEventArgs Properties

HitTestEventArgs.HitResult Property

Gets or sets the adjustment handle that has been hit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AdjustmentHandle HitResult { get; set; }
```

Visual Basic Copy Code

```
Public Property HitResult As AdjustmentHandle
```

Property Value

An instance of AdjustmentHandle -derived class, or **null** if there is no handle located at the specified position.

See Also

HitTestEventArgs Members
HitTestEventArgs Class
MindFusion.Diagramming Namespace

HitTestEventArgs.Item Property

Gets the item whose adjustment handles are being hit-tested.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem Item { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Item As DiagramItem
```

Property Value

An instance of the DiagramItem class.

See Also

HitTestEventArgs Members
HitTestEventArgs Class
MindFusion.Diagramming Namespace

HitTestEventArgs.MousePosition Property

Gets the current cursor position, expressed in document coordinates.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point MousePosition { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MousePosition As Point
```

Property Value

A **Microsoft.Maui.Graphics.Point** specifying the current mouse position.

See Also

HitTestEventArgs Members
HitTestEventArgs Class
MindFusion.Diagramming Namespace

HitTestPriority Enumeration

Defines values that specify the order of item hit-testing.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum HitTestPriority
```

Visual Basic Copy Code

```
Public Enum HitTestPriority
```

Members

	Member name	Description
	ZOrder	Test items in decreasing z-order, independently of their types.
	NodesBeforeLinks	Test nodes first, then links.

See Also

MindFusion.Diagramming Namespace
HitTestPriority Property

HitTestResult Enumeration

Represents the return value from a hit test.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum HitTestResult
```

Visual Basic Copy Code

```
Public Enum HitTestResult
```

Members

	Member name	Description
	Nothing	No recognizable element of the item has been hit.
	ScrollBar	A scrollbar has been hit.
	CollapseButton	A collapse/expand button has been hit.

See Also

MindFusion.Diagramming Namespace

HtmlBuilder Class

Provides methods to create and use HTML client side image maps.

Syntax

C# Copy Code

```
public class HtmlBuilder
```

Visual Basic Copy Code

```
Public Class HtmlBuilder
```

Remarks

If an exported diagram image is displayed in an HTML page, it is helpful to create an image map whose areas correspond to the items visible in the image. The CreateImageMap method generates a full <MAP> tag, including its <AREA> elements. CreateUseMapAttr method creates the code to reference a map from within an <IMG> tag, and also adds a WIDTH and HEIGHT attributes that define to the width and height of the exported image, measured in pixels.

Each <AREA> tag in the image map has its attributes set as follows:

- HREF is set to the HyperLink value of the diagram item or table's cell corresponding to the area;
- ALT is set to the ToolTip value of the node or table's cell corresponding to the area;
- TARGET is set to the value of HtmlBuilder's LinkTarget property.

Image map areas can be also generated for the [±] buttons displayed beside **Expandable** nodes. To enable that, assign an URL to the ExpandBtnHyperLink property. That URL can contain some format specifiers which are automatically replaced with the values of nodes' Tag or ZIndex properties. That lets you identify on the server side the node whose [±] button is clicked.

HtmlBuilder is most useful when it is used in conjunction with the WebChart library to create diagram images on the server, though it is available in the Windows Forms MvcDiagram control too.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.HtmlBuilder

See Also

HtmlBuilder Members
MindFusion.Diagramming Namespace

HtmlBuilder Members

Provides methods to create and use HTML client side image maps.

The following tables list the members exposed by the HtmlBuilder type.

Public Constructors

	Name	Description
	HtmlBuilder	Initializes a new instance of the **HtmlBuilder** class.

Public Properties

	Name	Description
	AreasForTableCells	Gets or sets a value indicating whether AREA tags are generated for table cells.
	AreasForTreeViewItems	Gets or sets a value indicating whether AREA tags are generated for TreeViewNode items.
	ExpandBtnHyperLink	Gets or sets the hyperlink assigned to AREA tags generated for the [±] buttons of **Expandable** nodes.
	LinkTarget	Gets or sets the value of the TARGET attribute of AREA tags.
	Scale	Gets or sets the scale factor for the generated image.
	ScrollDownHyperLink	Gets or sets the hyperlink assigned to AREA tags generated for the lower scroll arrow of tables.
	ScrollUpHyperLink	Gets or sets the hyperlink assigned to AREA tags generated for the upper scroll arrow of tables.

Public Methods

	Name	Description
	CreateImageHtml	Generates an image map HTML code.
	CreateImageMap	Overloaded.
	CreateUseMapAttr	Generates a USEMAP attribute to be used within an <IMG> tag.

Public Events

	Name	Description
	CreatingArea	Raised just before a new AREA tag is added to the image map.

See Also

HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder Constructor

Initializes a new instance of the HtmlBuilder class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HtmlBuilder (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram _
)
```

Parameters

diagram
The Diagram object whose contents will generate the image map.

See Also

HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder Methods

HtmlBuilder.CreateImageHtml Method

Generates an image map HTML code.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string CreateImageHtml (
    string htmlFileName,
    string htmlTitle,
    string imgLocalPath,
    string imgURL
)
```

Visual Basic Copy Code

```
Public Function CreateImageHtml( _
    htmlFileName As String, _
    htmlTitle As String, _
    imgLocalPath As String, _
    imgURL As String _
) As String
```

Parameters

htmlFileName
A string that represents the full name of the HTML file that is created.

htmlTitle
A string that represents the <TITLE> of the HTML file created.

imgLocalPath
A string that specifies the location of the image.

imgURL
A string that specifies the URL of the image at the server.

Return Value

A string with the HTML code of the created HTML file.

Remarks

The method also creates an HTML file with attributes specified as parameters.

See Also

HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder.CreateImageMap Method

Overload List

Name	Description
HtmlBuilder.CreateImageMap (String)	Creates an HTML code for a MAP tag.
HtmlBuilder.CreateImageMap (String, Rect)	Creates an HTML code for a MAP tag.

See Also

HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder.CreateImageMap Method (String)

Creates an HTML code for a MAP tag.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string CreateImageMap (
    string mapName
)
```

Visual Basic Copy Code

```
Public Function CreateImageMap( _
    mapName As String _
) As String
```

Parameters

mapName
A string that represents the NAME attribute in the <MAP> tag.

Return Value

A string value containing the generated HTML code.

Remarks

There are <AREA> tags generated for each node or link whose HyperLink is not an empty string. Outlines of items that cannot be represented exactly with standard <AREA> shapes are approximated to polygons. The HREF attribute of the area is set to the value of item's **HyperLink** property. The ALT attribute is set to the value of item's ToolTip property.

Example

The following HTML code:

HTML

```
<MAP NAME="diagram1">
    <AREA SHAPE ="RECT" COORDS="15,3,650,774" HREF="https://mindfusion.dev" ALT="">
```

```
    <AREA SHAPE ="POLY"
COORDS="198,43,195,48,185,52,170,55,151,57,130,58,109,57,90,55,75,52,65,48,62,43,65,38,75,34,90,31,10
HREF="" ALT="">
    <AREA SHAPE ="POLY" COORDS="96,85,164,85,198,96,198,119,164,130,96,130,62,119,62,96" HREF="https:
    <AREA SHAPE ="RECT" COORDS="60,151,196,196" HREF="" ALT="">
    <AREA SHAPE ="RECT" COORDS="60,230,196,287" HREF="" ALT="">
</MAP>
```

is generated from the following code:

C# Copy Code

```
HtmlBuilder hb = new HtmlBuilder(diagram);
string htmlCode = hb.CreateImageMap("diagram1");
```

Visual Basic Copy Code

```
Dim hb As HtmlBuilder = New HtmlBuilder(Diagram)
Dim htmlCode As String = hb.CreateImageMap("diagram1")
```

See Also

CreateImageMap Method Overload List
HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder.CreateImageMap Method (String, Rect)

Creates an HTML code for a MAP tag.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string CreateImageMap (
    string mapName,
    Rect viewport
)
```

Visual Basic Copy Code

```
Public Function CreateImageMap( _
    mapName As String, _
    viewport As Rect _
) As String
```

Parameters

mapName

A string that represents the NAME attribute in the <MAP> tag.

viewport

Specifies the diagram region for which to generate an image map.

Return Value

A string value containing the generated HTML code.

See Also

CreateImageMap Method Overload List
HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder.CreateUseMapAttr Method

Generates a USEMAP attribute to be used within an <IMG> tag.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string CreateUseMapAttr (
    string mapID
)
```

Visual Basic Copy Code

```
Public Function CreateUseMapAttr( _
    mapID As String _
) As String
```

Parameters

mapID
A string that represents the ID in the USEMAP attribute of the <IMG> HTML tag.

Return Value

A string that contains the generated HTML code for the USEMAP attribute.

Remarks

Use the same *mapID* as the one you have passed to CreateImageMap. The method generates also WIDTH and HEIGHT attributes, converting the diagram's MeasureUnit values to pixel values.

Example

The result of this method looks like:

HTML Copy Code

```
USEMAP="#diagram1" WIDTH="672" HEIGHT="808"
```

and is ready to be inserted in your <IMG> tag.

See Also

HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder Properties

HtmlBuilder.AreasForTableCells Property

Gets or sets a value indicating whether AREA tags are generated for table cells.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AreasForTableCells { get; set; }
```

Visual Basic Copy Code

```
Public Property AreasForTableCells As Boolean
```

Property Value

true to create AREA tags for table cells; otherwise, **false**. The default is **false**.

Remarks

If disabled, there is a single <AREA> created for each table, corresponding to the table's bounding rectangle. If enabled, each cell of a table has a corresponding <AREA> tag created and one <AREA> is created for the table's caption bar rectangle.

See Also

HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace
Table.Cell.HyperLink Property

HtmlBuilder.AreasForTreeViewItems Property

Gets or sets a value indicating whether AREA tags are generated for TreeViewNode items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AreasForTreeViewItems { get; set; }
```

Visual Basic Copy Code

```
Public Property AreasForTreeViewItems As Boolean
```

Property Value

true to create AREA tags for TreeViewNode items; otherwise, **false**. The default is **false**.

See Also

HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder.ExpandBtnHyperLink Property

Gets or sets the hyperlink assigned to AREA tags generated for the [±] buttons of **Expandable** nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ExpandBtnHyperLink { get; set; }
```

Visual Basic Copy Code

```
Public Property ExpandBtnHyperLink As String
```

Property Value

A string value containing an URL. The default is an empty string.

Remarks

If the value of this property is not an empty string, the HtmlBuilder creates <AREA> tags for the expand/collapse buttons displayed besides nodes. That lets you process requests to expand or collapse a tree branch on the server, or just use the [±] button as additional hot spots in the image map. The hyperlink string can contain any of the following format specifiers:

Specifier	Description
{0}	Replaced with the ZIndex of the node associated with the [±] button.
{1}	Replaced with the Tag of the node associated with the [±] button.

The **ZIndex** of a node can be used to directly access the node within the Items collection, it has the same value as the index of an item within that collection. The **Tag** can be used to find a node by calling the FindNode method.

Example

To have a JavaScript method called with node's Tag passed as parameter when a [±] button area is clicked, setup an **HtmlBuilder** instance as shown below:

C# Copy Code

```
HtmlBuilder hb = new HtmlBuilder(diagram);
hb.ExpandBtnHyperLink = "javascript: void processExpand({1})";
...
```

Visual Basic Copy Code

```
Dim hb As New HtmlBuilder(diagram)
hb.ExpandBtnHyperLink = "javascript: void processExpand({1})"
...
```

See Also

HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder.LinkTarget Property

Gets or sets the value of the TARGET attribute of AREA tags.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string LinkTarget { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkTarget As String
```

Property Value

A string containing the target frame name. The default is "_self".

Remarks

This property provides a value for the TARGET attribute of each <AREA> tag included in the image map. TARGET indicates the target window or frame where image area hyperlinks are opened. The target could be either a named frame or one of the following special values:

Value	Description
_blank	Loads the linked document into a new blank window.
_parent	Loads the linked document into current document's parent.

_self	Loads the linked document into the same window.
_top	Loads the linked document into the topmost browser window.

See Also

HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder.Scale Property

Gets or sets the scale factor for the generated image.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Scale { get; set; }
```

Visual Basic Copy Code

```
Public Property Scale As Double
```

Property Value

A **double** value that specifies the scale factor for the generated diagram image. The default is 100 (indicating no scale transformation).

See Also

HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder.ScrollDownHyperLink Property

Gets or sets the hyperlink assigned to AREA tags generated for the lower scroll arrow of tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ScrollDownHyperLink { get; set; }
```

Visual Basic Copy Code

```
Public Property ScrollDownHyperLink As String
```

Property Value

A string value containing an URL. The default is an empty string.

Remarks

If the value of this property is not an empty string, the HtmlBuilder creates <AREA> tags for the scroll-down button displayed for Scrollable tables. That lets you process requests to scroll the table rows on the server, or just use the scroll buttons as additional hot spots in the image map. If the URL contains the {0} format specifier, it is replaced with the ZIndex of the clicked table.

See Also

HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder.ScrollUpHyperLink Property

Gets or sets the hyperlink assigned to AREA tags generated for the upper scroll arrow of tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ScrollUpHyperLink { get; set; }
```

Visual Basic Copy Code

```
Public Property ScrollUpHyperLink As String
```

Property Value

A string value containing an URL. The default is an empty string.

Remarks

If the value of this property is not an empty string, the HtmlBuilder creates <AREA> tags for the scroll-down button displayed for Scrollable tables. That lets you process requests to scroll the table rows on the server, or just use the scroll buttons as additional hot spots in the image map. If the URL contains the {0} format specifier, it is replaced with the ZIndex of the clicked table.

See Also

HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HtmlBuilder Events

HtmlBuilder.CreatingArea Event

Raised just before a new AREA tag is added to the image map.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<AreaEventArgs> CreatingArea
```

Visual Basic Copy Code

```
Public Event CreatingArea As EventHandler(Of AreaEventArgs)
```

Event Data

CreatingArea event handlers receive an argument of type AreaEventArgs. The following **AreaEventArgs** members provide information relevant to the event:

Member name	Description

AreaTag	The HTML AREA tag created by HtmlBuilder.
Item	The diagram element represented by AreaTag in the image map.
Column	Contains the index of a column if the event is raised for a table cell.
Row	Contains the index of a row if the event is raised for a table cell.

Remarks

You could handle this event in order to insert additional attributes or **JavaScript** event handlers into the AREA tag. The tag string already ends with a closing ">" when the event is raised, so you must insert any additional attributes one character before the end of the tag string.

Example

The following code adds a JavaScript onClick event handler to the area tag:

C# Copy Code

```
private void htb_CreatingArea(object sender, AreaEventArgs e)
{
    string script = " onClick=\"alert('click click')\"";
    e.AreaTag = e.AreaTag.Insert(e.AreaTag.Length - 1, script);
}
```

Visual Basic Copy Code

```
Private Sub htb_CreatingArea(ByVal sender As Object, ByVal e As AreaEventArgs)

    Dim script As String = " onClick=" + Chr(34) + "alert('click click')" + Chr(34)
    e.AreaTag = e.AreaTag.Insert(e.AreaTag.Length - 1, script)

End Sub
```

See Also

HtmlBuilder Members
HtmlBuilder Class
MindFusion.Diagramming Namespace

HyperLinkStyle Enumeration

Specifies how various export components should treat the HyperLink property of items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum HyperLinkStyle
```

Visual Basic Copy Code

```
Public Enum HyperLinkStyle
```

Members

	Member name	Description
	None	Do not generate hyperlinks.

	Text	Generate hyperlinks for the text bounding rectangle.
	Shape	Generate hyperlinks that correspond to the item's shape.
	ShapeAndText	Generate hyperlinks for the item's shape and text bounding rectangle.

Remarks

At this time, members of this enumeration can be assigned to the HyperLinkStyle property of SvgExporter and PdfExporter.

See Also

MindFusion.Diagramming Namespace

IDiagramView Interface

Controls that display Diagram objects must implement this interface to provide screen resolution information to drawing methods.

Syntax

C# Copy Code

```
public interface IDiagramView : IResolution
```

Visual Basic Copy Code

```
Public Interface IDiagramView
    Implements IResolution
```

See Also

IDiagramView Members
MindFusion.Diagramming Namespace

IDiagramView Members

Controls that display Diagram objects must implement this interface to provide screen resolution information to drawing methods.

The following tables list the members exposed by the IDiagramView type.

Public Properties

	Name	Description
	Diagram	Gets the **Diagram** instance displayed by this control.

Public Methods

	Name	Description
	ZoomToRect	Zooms and scrolls the view to fit the specified document rectangle in the view's client area.

See Also

IDiagramView Interface
MindFusion.Diagramming Namespace

IDiagramView Methods

IDiagramView.ZoomToRect Method

Zooms and scrolls the view to fit the specified document rectangle in the view's client area.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
void ZoomToRect (
    Rect rect,
    bool centered
)
```

Visual Basic Copy Code

```
Sub ZoomToRect( _
    rect As Rect, _
    centered As Boolean _
)
```

Parameters

rect
A **Microsoft.Maui.Graphics.Rect** specifying the desired viewport's position and size.

centered
true to center the rectangle in viewport, or **false** to align it to top-left corner.

See Also

IDiagramView Members
IDiagramView Interface
MindFusion.Diagramming Namespace

IDiagramView Properties

IDiagramView.Diagram Property

Gets the Diagram instance displayed by this control.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
Diagram Diagram { get; }
```

Visual Basic Copy Code

```
ReadOnly Property Diagram As Diagram
```

Property Value

An instance of the Diagram class.

See Also

IDiagramView Members
IDiagramView Interface
MindFusion.Diagramming Namespace

IEffectsProvider Interface

Visual effects provider.

Syntax

C# Copy Code

```
public interface IEffectsProvider
```

Visual Basic Copy Code

```
Public Interface IEffectsProvider
```

Inheritors

MindFusion.Diagramming.EffectsProvider

See Also

IEffectsProvider Members
MindFusion.Diagramming Namespace

IEffectsProvider Members

Visual effects provider.

The following tables list the members exposed by the IEffectsProvider type.

Public Methods

	Name	Description
	GenerateEmboss	Generates an emboss mask for the shape defined by the specified path.
	GenerateShadow	Generates a shadow for the shape defined by the specified path.

See Also

IEffectsProvider Interface
MindFusion.Diagramming Namespace

IEffectsProvider Methods

IEffectsProvider.GenerateEmboss Method

Generates an emboss mask for the shape defined by the specified path.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
IImage GenerateEmboss (
    Diagram diagram,
    IGraphicsPath path
)
```

Visual Basic Copy Code

```
Function GenerateEmboss( _
    diagram As Diagram, _
    path As IGraphicsPath _
) As IImage
```

Parameters

diagram

A reference to a Diagram object containing the shape whose emboss mask is being generated.

path

A GraphicsPath that represents of the shape outline.

Return Value

A **Microsoft.Maui.Graphics.IImage** instance containing the generated emboss mask.

See Also

IEffectsProvider Members
IEffectsProvider Interface
MindFusion.Diagramming Namespace

IEffectsProvider.GenerateShadow Method

Generates a shadow for the shape defined by the specified path.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
IImage GenerateShadow (
    Diagram diagram,
    IGraphicsPath path,
    int dispersion,
    Brush brush
)
```

Visual Basic Copy Code

```
Function GenerateShadow( _
    diagram As Diagram, _
    path As IGraphicsPath, _
    dispersion As Integer, _
    brush As Brush _
) As IImage
```

Parameters

diagram

path

dispersion

brush

Return Value

See Also

IEffectsProvider Members
IEffectsProvider Interface
MindFusion.Diagramming Namespace

ILinkRouter Interface

Defines methods that should be implemented by a link-router component. The control calls them when a new link path should be set in response to changes in the diagram, such as adding or modifying nodes.

Syntax

C# Copy Code

```
public interface ILinkRouter
```

Visual Basic Copy Code

```
Public Interface ILinkRouter
```

Inheritors

MindFusion.Diagramming.CompositeRouter
MindFusion.Diagramming.GridRouter
MindFusion.Diagramming.PatternRouter
MindFusion.Diagramming.QuickRouter

See Also

ILinkRouter Members
MindFusion.Diagramming Namespace

ILinkRouter Members

Defines methods that should be implemented by a link-router component. The control calls them when a new link path should be set in response to changes in the diagram, such as adding or modifying nodes.

The following tables list the members exposed by the ILinkRouter type.

Public Properties

	Name	Description
	Diagram	Gets or sets the diagram this router is associated with.

Public Methods

	Name	Description
	RouteLink	Called when a single link must be routed.
	RouteLinks	Called when more than one link must be routed.

See Also

ILinkRouter Interface
MindFusion.Diagramming Namespace

ILinkRouter Methods

ILinkRouter.RouteLink Method

Called when a single link must be routed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
bool RouteLink (
    DiagramLink link,
    ConnectionPoint originConnection,
    ConnectionPoint destinationConnection,
    bool nowCreating
)
```

Visual Basic Copy Code

```
Function RouteLink( _
    link As DiagramLink, _
    originConnection As ConnectionPoint, _
    destinationConnection As ConnectionPoint, _
    nowCreating As Boolean _
) As Boolean
```

Parameters

link
A DiagramLink object representing the link.

originConnection
Represents connection to Origin node, potentially different from link's stored connection when modifying first control point.

destinationConnection
Represents connection to Destination node, potentially different from link's stored connection when modifying last control point.

nowCreating
Indicates whether link is currently being created.

Return Value

true if a path has been found, otherwise false.

See Also

ILinkRouter Members
ILinkRouter Interface
MindFusion.Diagramming Namespace

ILinkRouter.RouteLinks Method

Called when more than one link must be routed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
DiagramLinkCollection RouteLinks (
    DiagramLinkCollection links
)
```

Visual Basic Copy Code

```
Function RouteLinks( _
    links As DiagramLinkCollection _
) As DiagramLinkCollection
```

Parameters

links
A DiagramLinkCollection containing the links that should be routed.

Return Value

A list of links that were not routed successfully.

See Also

ILinkRouter Members
ILinkRouter Interface
MindFusion.Diagramming Namespace

ILinkRouter Properties

ILinkRouter.Diagram Property

Gets or sets the diagram this router is associated with.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
Diagram Diagram { get; set; }
```

Visual Basic Copy Code

```
Property Diagram As Diagram
```

Property Value

See Also

ILinkRouter Members
ILinkRouter Interface
MindFusion.Diagramming Namespace

IMeasureUnitConverter Interface

Provides means to convert between cross-type measure units.

Syntax

C# Copy Code

```
public interface IMeasureUnitConverter
```

Visual Basic Copy Code

```
Public Interface IMeasureUnitConverter
```

See Also

IMeasureUnitConverter Members
MindFusion.Diagramming Namespace

IMeasureUnitConverter Members

Provides means to convert between cross-type measure units.

The following tables list the members exposed by the IMeasureUnitConverter type.

Public Methods

	Name	Description
	CanConvertFrom	Returns a value indicating whether this converter can convert from the specified target unit to the specified source unit.
	CanConvertTo	Returns a value indicating whether this converter can convert from the specified source unit to the specified target unit.
	ConvertFrom	Converts a value from the specified source unit to the specified target unit.
	ConvertTo	Converts a value from the specified source unit to the specified target unit.

See Also

IMeasureUnitConverter Interface
MindFusion.Diagramming Namespace

IMeasureUnitConverter Methods

IMeasureUnitConverter.CanConvertFrom Method

Returns a value indicating whether this converter can convert from the specified target unit to the specified source unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
bool CanConvertFrom (
    MeasureUnit targetUnit,
    MeasureUnit sourceUnit
)
```

Visual Basic Copy Code

```
Function CanConvertFrom( _
    targetUnit As MeasureUnit, _
    sourceUnit As MeasureUnit _
) As Boolean
```

Parameters

targetUnit
The target MeasureUnit instance.

sourceUnit
The source **MeasureUnit** instance.

Return Value

true if conversion between the specified units is supported, or **false** otherwise.

See Also

IMeasureUnitConverter Members
IMeasureUnitConverter Interface
MindFusion.Diagramming Namespace

IMeasureUnitConverter.CanConvertTo Method

Returns a value indicating whether this converter can convert from the specified source unit to the specified target unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
bool CanConvertTo (
    MeasureUnit sourceUnit,
    MeasureUnit targetUnit
)
```

Visual Basic Copy Code

```
Function CanConvertTo( _
    sourceUnit As MeasureUnit, _
    targetUnit As MeasureUnit _
) As Boolean
```

Parameters

sourceUnit

The source MeasureUnit instance.

targetUnit

The target **MeasureUnit** instance.

Return Value

true if conversion between the specified units is supported, or **false** otherwise.

See Also

IMeasureUnitConverter Members
IMeasureUnitConverter Interface
MindFusion.Diagramming Namespace

IMeasureUnitConverter.ConvertFrom Method

Converts a value from the specified source unit to the specified target unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
double ConvertFrom (
    MeasureUnit targetUnit,
    MeasureUnit sourceUnit,
    double value,
    Axis axis,
    Object param
)
```

Visual Basic Copy Code

```
Function ConvertFrom( _
    targetUnit As MeasureUnit, _
    sourceUnit As MeasureUnit, _
    value As Double, _
    axis As Axis, _
    param As Object _
) As Double
```

Parameters

targetUnit

The target MeasureUnit instance.

sourceUnit

The source **MeasureUnit** instance.

value
A **double** value that should be converted.

axis
A member of the Axis enumeration specifying whether the value represents a horizontal or a vertical dimension.

param
An object providing device resolution information.

Return Value

The converted value.

Remarks

param can be set to either **Graphics**, IGraphics, **DiagramView** or IResolution object.

See Also

IMeasureUnitConverter Members
IMeasureUnitConverter Interface
MindFusion.Diagramming Namespace

IMeasureUnitConverter.ConvertTo Method

Converts a value from the specified source unit to the specified target unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
double ConvertTo (
    MeasureUnit sourceUnit,
    MeasureUnit targetUnit,
    double value,
    Axis axis,
    Object param
)
```

Visual Basic Copy Code

```
Function ConvertTo( _
    sourceUnit As MeasureUnit, _
    targetUnit As MeasureUnit, _
    value As Double, _
    axis As Axis, _
    param As Object _
) As Double
```

Parameters

sourceUnit
The source MeasureUnit instance.

targetUnit
The target **MeasureUnit** instance.

value
A **double** value that should be converted.

axis
A member of the Axis enumeration specifying whether the value represents a horizontal or a vertical dimension.

param
An object providing device resolution information.

Return Value

The converted value.

Remarks

param can be set to either **Graphics**, IGraphics, **DiagramView** or IResolution object.

See Also

IMeasureUnitConverter Members
IMeasureUnitConverter Interface
MindFusion.Diagramming Namespace

InplaceEditable Interface

Represents an in-place editable object.

Syntax

C# Copy Code

```
public interface InplaceEditable
```

Visual Basic Copy Code

```
Public Interface InplaceEditable
```

Inheritors

MindFusion.Diagramming.DiagramItem
MindFusion.Diagramming.ItemLabel
MindFusion.Diagramming.TableNode.Cell
MindFusion.Diagramming.TreeViewItem

See Also

InplaceEditable Members
MindFusion.Diagramming Namespace

InplaceEditable Members

Represents an in-place editable object.

The following tables list the members exposed by the InplaceEditable type.

Public Methods

	Name	Description
	GetEditRect	Returns the rectangle where the in-place editing control appears (in document coordinates).
	GetTextToEdit	Returns the text of the object which should be displayed initially when the object is in-place edited.
	SetEditedText	Sets the edited text to the object after the in-place editing operation has been completed successfully.

See Also

InplaceEditable Interface
MindFusion.Diagramming Namespace

InplaceEditable Methods

InplaceEditable.GetEditRect Method

Returns the rectangle where the in-place editing control appears (in document coordinates).

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
Rect GetEditRect (
    DiagramItem item,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Function GetEditRect( _
    item As DiagramItem, _
    mousePosition As Point _
) As Rect
```

Parameters

item
The item being edited.

mousePosition
A **Microsoft.Maui.Graphics.Point** instance specifying the mouse position.

Return Value

A **Microsoft.Maui.Graphics.Rect** that represents the destination rectangle of the in-place editing control.

See Also

InplaceEditable Members
InplaceEditable Interface
MindFusion.Diagramming Namespace

InplaceEditable.GetTextToEdit Method

Returns the text of the object which should be displayed initially when the object is in-place edited.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
string GetTextToEdit ()
```

Visual Basic Copy Code

```
Function GetTextToEdit () As String
```

Return Value

A string value that represents the initial contents of the in-place editing box.

See Also

InplaceEditable Members
InplaceEditable Interface
MindFusion.Diagramming Namespace

InplaceEditable.SetEditedText Method

Sets the edited text to the object after the in-place editing operation has been completed successfully.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
void SetEditedText (
    string newText
)
```

Visual Basic Copy Code

```
Sub SetEditedText( _
    newText As String _
)
```

Parameters

newText
The new text of the in-place edited object.

See Also

InplaceEditable Members
InplaceEditable Interface
MindFusion.Diagramming Namespace

InplaceEditableList Interface

Represents an object that can be edited via an inplace list control.

Syntax

C# Copy Code

```
public interface InplaceEditableList
```

Visual Basic Copy Code

```
Public Interface InplaceEditableList
```

Inheritors

MindFusion.Diagramming.PagedContainerNode

See Also

InplaceEditableList Members
MindFusion.Diagramming Namespace

InplaceEditableList Members

Represents an object that can be edited via an inplace list control.

The following tables list the members exposed by the InplaceEditableList type.

Public Methods

	Name	Description

	GetCurrentItem	Gets the index of the currently selected item.
	GetListItems	Gets the list of items to be displayed in the editor.
	SetCurrentItem	Sets the currently selected item by its index.

See Also

InplaceEditableList Interface
MindFusion.Diagramming Namespace

InplaceEditableList Methods

InplaceEditableList.GetCurrentItem Method

Gets the index of the currently selected item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
int GetCurrentItem ()
```

Visual Basic Copy Code

```
Function GetCurrentItem () As Integer
```

Return Value

See Also

InplaceEditableList Members
InplaceEditableList Interface
MindFusion.Diagramming Namespace

InplaceEditableList.GetListItems Method

Gets the list of items to be displayed in the editor.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
IList<string> GetListItems ()
```

Visual Basic Copy Code

```
Function GetListItems () As IList(Of String)
```

Return Value

A list of strings representing the items.

See Also

InplaceEditableList Members
InplaceEditableList Interface
MindFusion.Diagramming Namespace

InplaceEditableList.SetCurrentItem Method

Sets the currently selected item by its index.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
void SetCurrentItem (
    int index
)
```

Visual Basic Copy Code

```
Sub SetCurrentItem( _
    index As Integer _
)
```

Parameters

index
The index of the item to select.

See Also

InplaceEditableList Members
InplaceEditableList Interface
MindFusion.Diagramming Namespace

InstantiateItemEventArgs Class

Contains the arguments passed to the InstantiateItem event handlers.

Syntax

C# Copy Code

```
public class InstantiateItemEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class InstantiateItemEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.InstantiateItemEventArgs

See Also

InstantiateItemEventArgs Members
MindFusion.Diagramming Namespace

InstantiateItemEventArgs Members

Contains the arguments passed to the InstantiateItem event handlers.

The following tables list the members exposed by the InstantiateItemEventArgs type.

Public Constructors

	Name	Description
	InstantiateItemEventArgs	Initializes a new instance of the **InstantiateItemEventArgs** class.

Public Properties

	Name	Description
	Item	Gets or sets a reference to the created diagram item.
	ItemClass	Gets the item class identifier.

See Also

InstantiateItemEventArgs Class
MindFusion.Diagramming Namespace

InstantiateItemEventArgs Constructor

Initializes a new instance of the InstantiateItemEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public InstantiateItemEventArgs (
    string itemClass
)
```

Visual Basic Copy Code

```
Public New ( _
    itemClass As String _
)
```

Parameters

itemClass
The class string identifier specifying the type of the new item.

See Also

InstantiateItemEventArgs Class
MindFusion.Diagramming Namespace

InstantiateItemEventArgs Properties

InstantiateItemEventArgs.Item Property

Gets or sets a reference to the created diagram item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem Item { get; set; }
```

Visual Basic Copy Code

```
Public Property Item As DiagramItem
```

Property Value

An instance of the DiagramItem class.

See Also

InstantiateItemEventArgs Members
InstantiateItemEventArgs Class
MindFusion.Diagramming Namespace

InstantiateItemEventArgs.ItemClass Property

Gets the item class identifier.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ItemClass { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ItemClass As String
```

Property Value

A string value specifying the item class identifier.

See Also

InstantiateItemEventArgs Members
InstantiateItemEventArgs Class
MindFusion.Diagramming Namespace

InteractionState Class

Encapsulates the current state of user interaction with the control.

Syntax

C# Copy Code

```
public class InteractionState
```

Visual Basic Copy Code

```
Public Class InteractionState
```

Remarks

Return an instance of this class from the **StartDraw** method of a custom **Behavior** class.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.InteractionState

See Also

InteractionState Members
MindFusion.Diagramming Namespace

InteractionState Members

Encapsulates the current state of user interaction with the control.

The following tables list the members exposed by the InteractionState type.

Public Constructors

	Name	Description
	InteractionState	Initializes a new instance of the **InteractionState** class.

Public Properties

	Name	Description
	Action	Gets the type of Action currently performed by the user in the Diagram.
	AdjustmentHandle	Gets the adjustment handle being dragged by the user.
	CurrentItem	Gets a reference to the DiagramItem to be created or modified.
	CurrentPoint	Gets the current location of the mouse pointer.
	Diagram	Gets the current **Diagram** instance.
	InvalidRect	Gets or sets the area that should be repainted.
	LinkTarget	Gets or sets the current target node when a link is being modified.
	MouseMoved	Gets or sets a value indicating whether the mouse pointer has moved since the user pressed down the mouse button.
	SkipSavingState	Gets a value indicating whether diagram items should skip storing their state for subsequent cancel.
	StartPoint	Gets the mouse position where the interaction started.

Public Methods

	Name	Description
	Cancel	Cancels the user interaction.
	CancelNow	Instruct the interaction to terminate on the next pass.
	Complete	This method is invoked to complete the interaction, usually when the user releases the left mouse button.
	IsAllowed	Checks whether the current interaction can be completed at the specified point.
	IsCompleting	Returns whether the interaction has been completed successfully.
	SetHandleIndex	Sets the adjustment handle tracked by this object to the specified handle.
	SetItemDeleted	Notifies the interaction that its associated diagram item has been deleted.

	ShouldCancelDrag	Gets a value indicating whether the interaction should terminate on the next pass.
	Start	This method is invoked in the beginning of the interaction, usually when the user presses the left mouse button over a modification handle.
	Update	This method is invoked to update the interaction state, usually in response to the user moving the mouse after the modification has started.
	ValidateAndSetCursor	Sets the cursor depending on whether the interaction can be completed at the specified point or not.

See Also

InteractionState Class
MindFusion.Diagramming Namespace

InteractionState Constructor

Initializes a new instance of the InteractionState class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public InteractionState (
    DiagramItem currentItem,
    AdjustmentHandle adjustmentHandle,
    Action action
)
```

Visual Basic Copy Code

```
Public New ( _
    currentItem As DiagramItem, _
    adjustmentHandle As AdjustmentHandle, _
    action As Action _
)
```

Parameters

currentItem
The DiagramItem to be created or modified.

adjustmentHandle
The selection handle to be dragged with the mouse, defines what kind of modification to perform.

action
One of the Action enumeration values, specifying what kind of action to perform.

See Also

InteractionState Class
MindFusion.Diagramming Namespace

InteractionState Methods

InteractionState.Cancel Method

Cancels the user interaction.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Cancel (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public Sub Cancel( _
    diagram As Diagram _
)
```

Parameters

diagram
The Diagram, at which the action is performed.

Remarks

Call this method to cancel interaction, usually in response to the user clicking the right mouse button while dragging.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace
CancelNow

InteractionState.CancelNow Method

Instruct the interaction to terminate on the next pass.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void CancelNow ()
```

Visual Basic Copy Code

```
Public Sub CancelNow ()
```

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace
Cancel

InteractionState.Complete Method

This method is invoked to complete the interaction, usually when the user releases the left mouse button.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Complete (
    Point point,
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public Sub Complete( _
    point As Point, _
    diagram As Diagram _
)
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** instance indicating the location of the cursor, in document coordinates.

diagram
The Diagram, at which the action is performed.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace
IsCompleting

InteractionState.IsAllowed Method

Checks whether the current interaction can be completed at the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsAllowed (
    Point point
)
```

Visual Basic Copy Code

```
Public Function IsAllowed( _
    point As Point _
) As Boolean
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** indicating the location of the cursor, in document coordinates.

Return Value

true if the action can be completed at *point*; otherwise, **false**.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.IsCompleting Method

Returns whether the interaction has been completed successfully.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsCompleting ()
```

Visual Basic Copy Code

```
Public Function IsCompleting () As Boolean
```

Return Value

true if the interaction has been completed successfully; otherwise, **false**.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace
Complete

InteractionState.SetHandleIndex Method

Sets the adjustment handle tracked by this object to the specified handle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetHandleIndex (
    AdjustmentHandle newHandle
)
```

Visual Basic Copy Code

```
Public Sub SetHandleIndex( _
    newHandle As AdjustmentHandle _
)
```

Parameters

newHandle

An AdjustmentHandle instance representing the modified handle.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.SetItemDeleted Method

Notifies the interaction that its associated diagram item has been deleted.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetItemDeleted ()
```

Visual Basic

```
Public Sub SetItemDeleted ()
```

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.ShouldCancelDrag Method

Gets a value indicating whether the interaction should terminate on the next pass.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool ShouldCancelDrag ()
```

Visual Basic

```
Public Function ShouldCancelDrag () As Boolean
```

Return Value

true if the interaction should terminate; otherwise, **false**.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.Start Method

This method is invoked in the beginning of the interaction, usually when the user presses the left mouse button over a modification handle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void Start (
    Point point,
    Diagram diagram
)
```

Visual Basic

```
Public Sub Start( _
    point As Point, _
    diagram As Diagram _
)
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** indicating the location of the cursor, in document coordinates.

diagram
The Diagram, at which the interaction is performed.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.Update Method

This method is invoked to update the interaction state, usually in response to the user moving the mouse after the modification has started.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Update (
    Point point,
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public Function Update( _
    point As Point, _
    diagram As Diagram _
) As Boolean
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** indicating the location of the cursor, in document coordinates.

diagram
The Diagram, at which the interaction is performed.

Return Value

true if the update has been successful; otherwise, **false**.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.ValidateAndSetCursor Method

Sets the cursor depending on whether the interaction can be completed at the specified point or not.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CursorHint ValidateAndSetCursor (
    Point point,
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public Function ValidateAndSetCursor( _
    point As Point, _
```

```
    diagram As Diagram _
) As CursorHint
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** indicating the location of the cursor, in document coordinates.

diagram
The Diagram, at which the interaction is performed.

Return Value

One of the CursorHint enumeration values, which is used by **DiagramView** to set its **Cursor** property. The method might return CursorHint.DontChange and the cursor will be left as it is.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState Properties

InteractionState.Action Property

Gets the type of Action currently performed by the user in the Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Action Action { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Action As Action
```

Property Value

A member of the Action enumeration that specifies what kind of action the user currently performs on the Diagram.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.AdjustmentHandle Property

Gets the adjustment handle being dragged by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AdjustmentHandle AdjustmentHandle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property AdjustmentHandle As AdjustmentHandle
```

Property Value

An AdjustmentHandle object representing the dragged handle.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.CurrentItem Property

Gets a reference to the DiagramItem to be created or modified.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem CurrentItem { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property CurrentItem As DiagramItem
```

Property Value

An instance of the DiagramItem class.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.CurrentPoint Property

Gets the current location of the mouse pointer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point CurrentPoint { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property CurrentPoint As Point
```

Property Value

A **Microsoft.Maui.Graphics.Point** that specifies the current location of the cursor, in document coordinates.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.Diagram Property

Gets the current Diagram instance.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Diagram Diagram { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Diagram As Diagram
```

Property Value

An instance of the Diagram class.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.InvalidRect Property

Gets or sets the area that should be repainted.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect InvalidRect { get; set; }
```

Visual Basic Copy Code

```
Public Property InvalidRect As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** specifying a rectangular region of the diagram that should be repainted.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.LinkTarget Property

Gets or sets the current target node when a link is being modified.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode LinkTarget { get; set; }
```

Visual Basic

```
Public Property LinkTarget As DiagramNode
```

Property Value

A DiagramNode that the currently modified DiagramLink should connect to if the interaction completes at the current mouse position.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.MouseMoved Property

Gets or sets a value indicating whether the mouse pointer has moved since the user pressed down the mouse button.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool MouseMoved { get; set; }
```

Visual Basic

```
Public Property MouseMoved As Boolean
```

Property Value

true is the mouse has moved; otherwise, **false**.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.SkipSavingState Property

Gets a value indicating whether diagram items should skip storing their state for subsequent cancel.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool SkipSavingState { get; }
```

Visual Basic

```
Public ReadOnly Property SkipSavingState As Boolean
```

Property Value

A boolean value.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

InteractionState.StartPoint Property

Gets the mouse position where the interaction started.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point StartPoint { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property StartPoint As Point
```

Property Value

A **Microsoft.Maui.Graphics.Point** specifying the mouse position where the user started to draw or modify an item.

See Also

InteractionState Members
InteractionState Class
MindFusion.Diagramming Namespace

IProperty Interface

Represents a property that can have an undefined value.

Syntax

C# Copy Code

```
public interface IProperty
```

Visual Basic Copy Code

```
Public Interface IProperty
```

See Also

IProperty Members
MindFusion.Diagramming Namespace

IProperty Members

Represents a property that can have an undefined value.

The following tables list the members exposed by the IProperty type.

Public Properties

	Name	Description
	HasValue	Gets a value indicating whether the property has a value.
	Name	Gets the name of the property.

Public Methods

	Name	Description

	GetValue	Gets the value of the property.
	SetValue	Sets the value of the property.
	Unset	Clears the value of the property.

See Also

IProperty Interface
MindFusion.Diagramming Namespace

IProperty Methods

IProperty.GetValue Method

Gets the value of the property.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
Object GetValue ()
```

Visual Basic Copy Code

```
Function GetValue () As Object
```

Return Value

See Also

IProperty Members
IProperty Interface
MindFusion.Diagramming Namespace

IProperty.SetValue Method

Sets the value of the property.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
void SetValue (
    Object value
)
```

Visual Basic Copy Code

```
Sub SetValue( _
    value As Object _
)
```

Parameters

value

See Also

IProperty Members
IProperty Interface
MindFusion.Diagramming Namespace

IProperty.Unset Method

Clears the value of the property.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
void Unset ()
```

Visual Basic Copy Code

```
Sub Unset ()
```

See Also

IProperty Members
IProperty Interface
MindFusion.Diagramming Namespace

IProperty Properties

IProperty.HasValue Property

Gets a value indicating whether the property has a value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
bool HasValue { get; }
```

Visual Basic Copy Code

```
ReadOnly Property HasValue As Boolean
```

Property Value

See Also

IProperty Members
IProperty Interface
MindFusion.Diagramming Namespace

IProperty.Name Property

Gets the name of the property.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
string Name { get; }
```

Visual Basic Copy Code

```
ReadOnly Property Name As String
```

Property Value

See Also

IProperty Members
IProperty Interface
MindFusion.Diagramming Namespace

IResolution Interface

Provides information about device resolution.

Syntax

C# Copy Code

```
public interface IResolution
```

Visual Basic Copy Code

```
Public Interface IResolution
```

Inheritors

MindFusion.Diagramming.IDiagramView

See Also

IResolution Members
MindFusion.Diagramming Namespace

IResolution Members

Provides information about device resolution.

The following tables list the members exposed by the IResolution type.

Public Properties

	Name	Description
	DpiX	Gets the horizontal resolution of the device.
	DpiY	Gets the vertical resolution of the device.

See Also

IResolution Interface
MindFusion.Diagramming Namespace

IResolution Properties

IResolution.DpiX Property

Gets the horizontal resolution of the device.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
double DpiX { get; }
```

Visual Basic Copy Code

```
ReadOnly Property DpiX As Double
```

Property Value

A **double** value specifying horizontal dots-per-inch resolution.

See Also

IResolution Members
IResolution Interface
MindFusion.Diagramming Namespace

IResolution.DpiY Property

Gets the vertical resolution of the device.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
double DpiY { get; }
```

Visual Basic Copy Code

```
ReadOnly Property DpiY As Double
```

Property Value

A **double** value specifying vertical dots-per-inch resolution.

See Also

IResolution Members
IResolution Interface
MindFusion.Diagramming Namespace

ItemCollectionBase(T) Class

A base collection class for the diagram item collections.

Syntax

C# Copy Code

```
public abstract class ItemCollectionBase<T> : CollectionBase<T> where T : DiagramItem
```

Visual Basic Copy Code

```
Public MustInherit Class ItemCollectionBase(Of T As DiagramItem)
    Inherits CollectionBase(Of T)
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.CollectionBase(T)
    MindFusion.Diagramming.ItemCollectionBase(T)
      MindFusion.Diagramming.DiagramItemCollection
      MindFusion.Diagramming.DiagramLinkCollection
      MindFusion.Diagramming.DiagramNodeCollection

See Also

ItemCollectionBase(T) Members
MindFusion.Diagramming Namespace

ItemCollectionBase(T) Members

A base collection class for the diagram item collections.

The following tables list the members exposed by the ItemCollectionBase(T) type.

Protected Fields

	Name	Description
	InnerList	Gets a **List<T>** containing the list of elements in the collection instance. (Inherited from CollectionBase.)
	itemEventArgs	A ItemEventArgs instance used for passing arguments to the event handlers for the events of this collection. Used to avoid creating many small objects that would burden the garbage collector.
	notifyParent	If enabled, the collection raises events to let their containing objects handle the addition or removal of items.

Public Properties

	Name	Description
	Count	Gets the number of elements actually contained in the collection. (Inherited from CollectionBase.)
	IsReadOnly	Gets a value indicating whether the collection is read-only. (Inherited from CollectionBase.)
	Item	Gets or sets the element at the specified index. (Inherited from CollectionBase.)

Public Methods

	Name	Description
	Add	Adds the specified diagram item to the collection.
	Clear	Removes all diagram items from the collection.
	Contains	Determines whether an element is in the collection. (Inherited from CollectionBase.)
	CopyTo	Copies the entire collection to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from CollectionBase.)

	GetAt	Gets the diagram item at the specified position.
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from CollectionBase.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire collection. (Inherited from CollectionBase.)
	Insert	Inserts the specified diagram item into the collection at the specified position.
	Remove	Removes the diagram item from the collection.
	RemoveAt	Removes the diagram item from the collection.
	Sort	Overloaded. (Inherited from CollectionBase.)

Protected Methods

	Name	Description
	RaiseAdding	Raises the **Adding** event.
	RaiseInserting	Raises the **Inserting** event.
	RaiseRemoving	Raises the **Removing** event.

See Also

ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemCollectionBase(T) Fields

ItemCollectionBase(T).itemEventArgs Field

A ItemEventArgs instance used for passing arguments to the event handlers for the events of this collection. Used to avoid creating many small objects that would burden the garbage collector.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected ItemEventArgs itemEventArgs
```

Visual Basic Copy Code

```
Protected itemEventArgs As ItemEventArgs
```

See Also

ItemCollectionBase(T) Members
ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemCollectionBase(T).notifyParent Field

If enabled, the collection raises events to let their containing objects handle the addition or removal of items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected bool notifyParent
```

Visual Basic Copy Code

```
Protected notifyParent As Boolean
```

See Also

ItemCollectionBase(T) Members
ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemCollectionBase(T) Methods

ItemCollectionBase(T).Add Method

Adds the specified diagram item to the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Add (
    T item
)
```

Visual Basic Copy Code

```
Public Overrides Sub Add( _
    item As T _
)
```

Parameters

item
The DiagramItem object to add to the collection.

See Also

ItemCollectionBase(T) Members
ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemCollectionBase(T).Clear Method

Removes all diagram items from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Clear ()
```

Visual Basic Copy Code

```
Public Overrides Sub Clear ()
```

See Also

ItemCollectionBase(T) Members
ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemCollectionBase(T).GetAt Method

Gets the diagram item at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem GetAt (
    int index
)
```

Visual Basic Copy Code

```
Public Function GetAt( _
    index As Integer _
) As DiagramItem
```

Parameters

index
The zero-based index of the item to retrieve.

Return Value

An instance of the DiagramItem class that represents the item at the specified position.

See Also

ItemCollectionBase(T) Members
ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemCollectionBase(T).Insert Method

Inserts the specified diagram item into the collection at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Insert (
    int index,
    T item
)
```

Visual Basic Copy Code

```
Public Overrides Sub Insert( _
    index As Integer, _
    item As T _
)
```

Parameters

index
The zero-based index at which to insert the item.

item
The DiagramItem object to add to the collection.

See Also

ItemCollectionBase(T) Members
ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemCollectionBase(T).RaiseAdding Method

Raises the **Adding** event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void RaiseAdding (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Protected Sub RaiseAdding( _
    item As DiagramItem _
)
```

Parameters

item
The diagram item that is being added.

See Also

ItemCollectionBase(T) Members
ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemCollectionBase(T).RaiseInserting Method

Raises the **Inserting** event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void RaiseInserting (
    DiagramItem item,
    int index
)
```

Visual Basic

```
Protected Sub RaiseInserting( _
    item As DiagramItem, _
    index As Integer _
)
```

Parameters

item

The diagram item that is being inserted.

index

Index of the new item.

See Also

ItemCollectionBase(T) Members
ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemCollectionBase(T).RaiseRemoving Method

Raises the **Removing** event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected void RaiseRemoving (
    DiagramItem item
)
```

Visual Basic

```
Protected Sub RaiseRemoving( _
    item As DiagramItem _
)
```

Parameters

item

The diagram item that is being removed.

See Also

ItemCollectionBase(T) Members
ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemCollectionBase(T).Remove Method

Removes the diagram item from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override bool Remove (
    T item
)
```

Visual Basic

```
Public Overrides Function Remove( _
    item As T _
) As Boolean
```

Parameters

item
The DiagramItem object to remove from the collection.

Return Value

See Also

ItemCollectionBase(T) Members
ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemCollectionBase(T).RemoveAt Method

Removes the diagram item from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override void RemoveAt (
    int index
)
```

Visual Basic

```
Public Overrides Sub RemoveAt( _
    index As Integer _
)
```

Parameters

index
Index of the object that should be removed from this collection.

See Also

ItemCollectionBase(T) Members
ItemCollectionBase(T) Class
MindFusion.Diagramming Namespace

ItemEventArgs Class

Contains the arguments passed to item instantiation event handlers.

Syntax

C#

```
public class ItemEventArgs : EventArgs
```

Visual Basic

```
Public Class ItemEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.ItemEventArgs
      MindFusion.Diagramming.LinkEventArgs
      MindFusion.Diagramming.NodeEventArgs

See Also

ItemEventArgs Members
MindFusion.Diagramming Namespace

ItemEventArgs Members

Contains the arguments passed to item instantiation event handlers.

The following tables list the members exposed by the ItemEventArgs type.

Public Constructors

	Name	Description
	ItemEventArgs	Initializes a new instance of the **ItemEventArgs** class.

Public Properties

	Name	Description
	Index	Gets an integer index related to the event.
	Item	Gets the item for which the event is raised.

See Also

ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemEventArgs Constructor

Initializes a new instance of the ItemEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ItemEventArgs (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public New ( _
    item As DiagramItem _
)
```

Parameters

item

The DiagramItem related to the event.

See Also

ItemEventArgs Properties

ItemEventArgs.Index Property

Gets an integer index related to the event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Index { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Index As Integer
```

Property Value

An integer value specifying the item's position within its containing collection.

See Also

ItemEventArgs Members
ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemEventArgs.Item Property

Gets the item for which the event is raised.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem Item { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Item As DiagramItem
```

Property Value

An instance of the DiagramItem class.

See Also

ItemEventArgs Members
ItemEventArgs Class
MindFusion.Diagramming Namespace

ItemLabel Class

The base class for node and link labels.

Syntax

C# Copy Code

```
public class ItemLabel : InplaceEditable
```

Visual Basic Copy Code

```
Public Class ItemLabel
    Implements InplaceEditable
```

Remarks

This class represents supplementary labels displayed by an item, in addition to its main Text property. Call the AddLabel method to create a label, and customize it through various members of type-specific LinkLabel and NodeLabel classes.

A label can also display an image assigned to the Image or ImageUrl property. This lets you draw multiple icons per node, or along a link's geometry.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.ItemLabel
MindFusion.Diagramming.LinkLabel
MindFusion.Diagramming.NodeLabel
MindFusion.Diagramming.LinkLabel
MindFusion.Diagramming.NodeLabel

See Also

ItemLabel Members
MindFusion.Diagramming Namespace

ItemLabel Members

The base class for node and link labels.

The following tables list the members exposed by the ItemLabel type.

Public Constructors

	Name	Description
	ItemLabel	Overloaded.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to draw the label's background.
	ContentPadding	Gets or sets the distance between image and text when both are specified.
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled.
	EnableWrap	Gets or sets a value indicating whether the label's text can be wrapped.
	Font	Gets or sets the label's font.
	HorizontalAlign	Gets or sets the label's horizontal alignment.
	HorizontalOffset	Gets or sets the label's horizontal offset from its pin point.
	Image	Gets or sets the image displayed by this label.
	ImageAlign	Gets or sets the alignment of label's image relative to text.
	ImageSize	Gets or sets a size that overrides the native image size.

	ImageUrl	Gets or sets URL of the image to display inside this label.
	Index	Gets or sets the index of an edge or point this label is relative to.
	Item	Gets or sets the DiagramItem that displays this label.
	Locked	Gets or sets a value indicating whether users are allowed to edit this label.
	Margin	Gets or sets the label's margin.
	MaxWidth	Gets or sets the maximum width of the label.
	Text	Gets or sets the label's text.
	TextBrush	Gets or sets the brush used to draw the label's text.
	VerticalAlign	Gets or sets the label's vertical alignment.
	VerticalOffset	Gets or sets the label's vertical offset from its pin point.

Public Methods

	Name	Description
	Cancel	Called when drag operation is canceled.
	Draw	Draws the label on the specified IGraphics surface.
	GetEditRect	InplaceEditable.GetEditRect implementation.
	GetTextToEdit	InplaceEditable.GetTextToEdit implementation.
	LoadFromXml	Loads the label from the specified XML element.
	OnMouseDown	Called when the left mouse button is pressed while the pointer is over this label.
	OnMouseMove	Called when the mouse is moved while this label tracks the mouse events.
	OnMouseUp	Called when the left mouse button is released while this label tracks the mouse events.
	SaveToXml	Saves the label into the specified XML element.
	SetEditedText	InplaceEditable.SetEditedText implementation.
	SetMouseCursor	Indicates what cursor to display for interactive labels.
	UpdateValue	Called while drag operation is in progress.

Protected Methods

	Name	Description
	LoadFromJson	Loads the label from a JsonObject.
	SaveToJson	Saves the label into a **JsonObject**.

See Also

ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel Constructor

Overload List

Name	Description
ItemLabel ()	Initializes a new instance of the ItemLabel class.
ItemLabel (DiagramItem)	Initializes a new instance of the **ItemLabel** class.

See Also

ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel Constructor ()

Initializes a new instance of the ItemLabel class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ItemLabel ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ItemLabel Constructor Overload List
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel Constructor (DiagramItem)

Initializes a new instance of the ItemLabel class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ItemLabel (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public New ( _
    item As DiagramItem _
)
```

Parameters

item
The DiagramItem that displays the label.

See Also

ItemLabel Constructor Overload List
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel Methods

ItemLabel.Cancel Method

Called when drag operation is canceled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void Cancel ()
```

Visual Basic Copy Code

```
Public Overridable Sub Cancel ()
```

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.Draw Method

Draws the label on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void Draw (
    IGraphics graphics
)
```

Visual Basic Copy Code

```
Public Overridable Sub Draw( _
    graphics As IGraphics _
)
```

Parameters

graphics
An IGraphics object where the label should be rendered.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.GetEditRect Method

InplaceEditable.GetEditRect implementation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect GetEditRect (
    DiagramItem item,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public Function GetEditRect( _
    item As DiagramItem, _
    mousePosition As Point _
) As Rect
```

Parameters

item

The item being edited.

mousePosition

A **Microsoft.Maui.Graphics.Point** instance specifying the mouse position.

Return Value

A **Microsoft.Maui.Graphics.Rect** specifying position of in-place edit control.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.GetTextToEdit Method

InplaceEditable.GetTextToEdit implementation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string GetTextToEdit ()
```

Visual Basic Copy Code

```
Public Function GetTextToEdit () As String
```

Return Value

A string value containing label's text.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.LoadFromJson Method

Loads the label from a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject containing the label's data.

context

A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.LoadFromXml Method

Loads the label from the specified XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Overridable Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** instance.

context

An XmlPersistContext instance.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.OnMouseDown Method

Called when the left mouse button is pressed while the pointer is over this label.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void OnMouseDown (
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public Overridable Sub OnMouseDown( _
    mousePosition As Point _
)
```

Parameters

mousePosition
A **Microsoft.Maui.Graphics.Point** that specifies the current mouse position.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.OnMouseMove Method

Called when the mouse is moved while this label tracks the mouse events.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void OnMouseMove (
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public Overridable Sub OnMouseMove( _
    mousePosition As Point _
)
```

Parameters

mousePosition
A **Microsoft.Maui.Graphics.Point** that specifies the current mouse position.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.OnMouseUp Method

Called when the left mouse button is released while this label tracks the mouse events.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void OnMouseUp (
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public Overridable Sub OnMouseUp( _
    mousePosition As Point _
)
```

Parameters

mousePosition
A **Microsoft.Maui.Graphics.Point** that specifies the current mouse position.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.SaveToJson Method

Saves the label into a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject the label's data should be stored into.

context
A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.SaveToXml Method

Saves the label into the specified XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Overridable Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
A **XmlElement** instance.

context
A XmlPersistContext instance.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.SetEditedText Method

InplaceEditable.SetEditedText implementation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetEditedText (
    string newText
)
```

Visual Basic Copy Code

```
Public Sub SetEditedText( _
    newText As String _
)
```

Parameters

newText
The new text of the in-place edited object.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.SetMouseCursor Method

Indicates what cursor to display for interactive labels.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CursorHint SetMouseCursor (
    Point point
)
```

Visual Basic Copy Code

```
Public Function SetMouseCursor( _
    point As Point _
) As CursorHint
```

Parameters

point
Specifies mouse pointer position

Return Value

A member of the CursorHint enumeration.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.UpdateValue Method

Called while drag operation is in progress.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public virtual void UpdateValue (
    Point current
)
```

Visual Basic Copy Code

```
Public Overridable Sub UpdateValue( _
    current As Point _
)
```

Parameters

current
A **Microsoft.Maui.Graphics.Point** specifying current position of mouse pointer.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel Properties

ItemLabel.Brush Property

Gets or sets the brush used to draw the label's background.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush Brush { get; set; }
```

Visual Basic Copy Code

```
Public Property Brush As Brush
```

Property Value

A Brush used to paint the label's background. The default value is **null**.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.ContentPadding Property

Gets or sets the distance between image and text when both are specified.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Nullable<Double> ContentPadding { get; set; }
```

Visual Basic Copy Code

```
Public Property ContentPadding As Nullable(Of Double)
```

Property Value

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.EnableStyledText Property

Gets or sets a value indicating whether styled text rendering is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableStyledText { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableStyledText As Boolean
```

Property Value

true to parse the label's text in order to find formatting tags, or **false** otherwise.

Remarks

Styled text allows applying various attributes to the characters it contains. Attributes are specified using HTML-like formatting tags embedded in the raw text. Formatting tags are listed and described in the **Text Attributes and Formatting** section of this help file.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace
Text Property

ItemLabel.EnableWrap Property

Gets or sets a value indicating whether the label's text can be wrapped.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableWrap { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableWrap As Boolean
```

Property Value

true to enable text wrapping, or **false** otherwise.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.Font Property

Gets or sets the label's font.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Font Font { get; set; }
```

Visual Basic Copy Code

```
Public Property Font As Font
```

Property Value

A Font instance.

Remarks

This property can be **null** (**Nothing** in Visual Basic) in which case the label will use the Font of its parent item.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace
Text Property
TextBrush Property

ItemLabel.HorizontalAlign Property

Gets or sets the label's horizontal alignment.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringAlignment HorizontalAlign { get; set; }
```

Visual Basic Copy Code

```
Public Property HorizontalAlign As StringAlignment
```

Property Value

A member of the StringAlignment enumeration. The default is **Center**.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.HorizontalOffset Property

Gets or sets the label's horizontal offset from its pin point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double HorizontalOffset { get; set; }
```

Visual Basic Copy Code

```
Public Property HorizontalOffset As Double
```

Property Value

A **double** value specifying distance.

See Also

ItemLabel.Image Property

Gets or sets the image displayed by this label.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage Image { get; set; }
```

Visual Basic Copy Code

```
Public Property Image As IImage
```

Property Value

A **Microsoft.Maui.Graphics.IImage** instance. The default value is **null**.

Remarks

If a label displays both **Image** and Text, the image position relative to text depends on the ImageAlign property.

If the label is only intended to display an icon image and no text, you could also set its Locked property to prevent users from entering text.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace
ImageSize Property
ImageUrl Property

ItemLabel.ImageAlign Property

Gets or sets the alignment of label's image relative to text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ImageAlign ImageAlign { get; set; }
```

Visual Basic Copy Code

```
Public Property ImageAlign As ImageAlign
```

Property Value

A member of the ImageAlign enumeration. The default value is **MiddleLeft**.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace
Image Property
ImageUrl Property

ItemLabel.ImageSize Property

Gets or sets a size that overrides the native image size.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size ImageSize { get; set; }
```

Visual Basic Copy Code

```
Public Property ImageSize As Size
```

Property Value

A **Microsoft.Maui.Graphics.Size** instance. The default value is **Empty**.

Remarks

By default, Image is rendered using bitmap's intrinsic size. Set this value to increase or reduce the size of the rendered image.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace
Image Property
ImageUrl Property

ItemLabel.ImageUrl Property

Gets or sets URL of the image to display inside this label.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ImageUrl { get; set; }
```

Visual Basic Copy Code

```
Public Property ImageUrl As String
```

Property Value

A URL string that identifies the image file location.

Remarks

If a label displays both image and Text, the image position relative to text depends on the ImageAlign property.

If the label is only intended to display an icon image and no text, you could also set its Locked property to prevent users from entering text.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace
Image Property

ItemLabel.Index Property

Gets or sets the index of an edge or point this label is relative to.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Index { get; set; }
```

Visual Basic Copy Code

```
Public Property Index As Integer
```

Property Value

An integer index.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.Item Property

Gets or sets the DiagramItem that displays this label.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem Item { get; set; }
```

Visual Basic Copy Code

```
Public Property Item As DiagramItem
```

Property Value

A DiagramItem reference.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.Locked Property

Gets or sets a value indicating whether users are allowed to edit this label.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Locked { get; set; }
```

Visual Basic

```
Public Property Locked As Boolean
```

Property Value

true to prevent users from editing this label, or **false** otherwise.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.Margin Property

Gets or sets the label's margin.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Thickness Margin { get; set; }
```

Visual Basic

```
Public Property Margin As Thickness
```

Property Value

A Thickness instance specifying the size of text margins.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.MaxWidth Property

Gets or sets the maximum width of the label.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double MaxWidth { get; set; }
```

Visual Basic

```
Public Property MaxWidth As Double
```

Property Value

A **double** value.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.Text Property

Gets or sets the label's text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Text { get; set; }
```

Visual Basic Copy Code

```
Public Property Text As String
```

Property Value

A string containing label's text.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace
Font Property
EnableStyledText Property

ItemLabel.TextBrush Property

Gets or sets the brush used to draw the label's text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush TextBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property TextBrush As Brush
```

Property Value

An instance of a Brush -derived class.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.VerticalAlign Property

Gets or sets the label's vertical alignment.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringAlignment VerticalAlign { get; set; }
```

Visual Basic Copy Code

```
Public Property VerticalAlign As StringAlignment
```

Property Value

A member of the StringAlignment enumeration. The default is **Center**.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

ItemLabel.VerticalOffset Property

Gets or sets the label's vertical offset from its pin point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double VerticalOffset { get; set; }
```

Visual Basic Copy Code

```
Public Property VerticalOffset As Double
```

Property Value

A **double** value specifying distance.

See Also

ItemLabel Members
ItemLabel Class
MindFusion.Diagramming Namespace

JsonPersistContext Class

Contains methods that can help you serialize and deserialize MindFusion.Diagramming items to and from JSON.

Syntax

C# Copy Code

```
public class JsonPersistContext : JsonContext
```

Visual Basic Copy Code

```
Public Class JsonPersistContext
    Inherits JsonContext
```

Inheritance Hierarchy

System.Object
  MindFusion.Json.JsonContext
    MindFusion.Diagramming.JsonPersistContext

See Also

JsonPersistContext Members

Contains methods that can help you serialize and deserialize MindFusion.Diagramming items to and from JSON.

The following tables list the members exposed by the JsonPersistContext type.

Public Constructors

	Name	Description
	JsonPersistContext	Overloaded.

Public Fields

	Name	Description
	LatestFormat	Specifies the latest file format revision number.

Protected Fields

	Name	Description
	brushes	A list of all brushes. (Inherited from JsonContext.)
	inplaceImages	Specifies whether images are encoded in place. (Inherited from JsonContext.)
	inplaceResources	Specifies whether resource accumulation is disabled. (Inherited from JsonContext.)

Public Properties

	Name	Description
	Diagram	Gets the underlying diagram object.
	FileVersion	Gets the format revision number for the file being currently serialized.

Public Methods

	Name	Description
	InitItem	
	LoadItem	Reads a diagram item from the specified JsonObject.
	Parse	Deserializes the specified JSON formatted string to a JsonObject. (Inherited from JsonContext.)
	ReadAnchorPattern	Reads an AnchorPattern object.
	ReadBrush	Overloaded. (Inherited from JsonContext.)
	ReadBrushColor	Reads a Brush object with the specified color. (Inherited from JsonContext.)
	ReadBrushes	Reads all brushes. (Inherited from JsonContext.)
	ReadColor	Overloaded. (Inherited from JsonContext.)
	ReadColorBlend	Overloaded. (Inherited from JsonContext.)
	ReadComposite	Reads the properties of a CompositeNode, which are marked with the AutoJson attribute.

	ReadEffect	Reads a NodeEffect object.
	ReadEnum	Reads an enum value. (Inherited from JsonContext.)
	ReadFloatArray	Reads a float array. (Inherited from JsonContext.)
	ReadFont	Reads a Font object.
	ReadImage	Reads an Image object. (Inherited from JsonContext.)
	ReadImages	Reads all images. (Inherited from JsonContext.)
	ReadItemId	Reads an item reference.
	ReadMap	Reads a MapContent object.
	ReadObject	Reads an object. (Inherited from JsonContext.)
	ReadPen	Overloaded. (Inherited from JsonContext.)
	ReadPointF	Reads a **Point** object. (Inherited from JsonContext.)
	ReadPointString	Reads a Point string. (Inherited from JsonContext.)
	ReadRectangleF	Reads a Rect object. (Inherited from JsonContext.)
	ReadShape	Reads a Shape object.
	ReadSizeF	Reads a SizeF object. (Inherited from JsonContext.)
	ReadStringFormat	Reads a StringFormat object. (Inherited from JsonContext.)
	ReadStyle	Reads a Style object.
	ReadSvg	Reads a SVG objec.
	ReadTag	Reads the specified tag for the specified object.
	ReadTheme	Reads a Theme object.
	ReadThickness	Reads a Thickness object.
	SaveItem	Saves the specified diagram item to a JsonObject.
	Serialize	Serializes the specified JsonObject to a JSON formatted string. (Inherited from JsonContext.)
	WriteAnchorPattern	Writes an AnchorPattern object.
	WriteBrush	Writes a MindFusion.Drawing.Brush object. (Inherited from JsonContext.)
	WriteBrushColor	Writes the color of the specified brush. (Inherited from JsonContext.)
	WriteBrushes	Writes all currently accumulated brushes. (Inherited from JsonContext.)
	WriteColor	Writes a Color object. (Inherited from JsonContext.)
	WriteColorBlend	Overloaded. (Inherited from JsonContext.)
	WriteComposite	Writes the properties of a CompositeNode, which are marked with the AutoJson attribute.
	WriteEffect	Writes a NodeEffect object.
	WriteEnum	Writes an enum value. (Inherited from JsonContext.)
	WriteFont	Writes a Font object.
	WriteHandlesStyle	Writes a HandlesVisualStyle object.
	WriteImage	Writes an Image object. (Inherited from JsonContext.)
	WriteImageAsync	Writes an Image object. (Inherited from JsonContext.)

	WriteImages	Writes all currently accumulated images. (Inherited from JsonContext.)
	WriteImagesAsync	Writes all currently accumulated images asynchronously. (Inherited from JsonContext.)
	WriteItemId	Writes a reference to the specified item.
	WriteItemType	Writes an item type.
	WriteMap	Writes a MapContent object.
	WriteObject	Writes an object. (Inherited from JsonContext.)
	WritePen	Writes a MindFusion.Drawing.Pen object. (Inherited from JsonContext.)
	WritePointF	Writes a **Point** object. (Inherited from JsonContext.)
	WritePointString	Writes a Point string. (Inherited from JsonContext.)
	WriteRectangleF	Writes a Rect object. (Inherited from JsonContext.)
	WriteShape	Writes a Shape object.
	WriteSizeF	Writes a SizeF object. (Inherited from JsonContext.)
	WriteStringFormat	Writes a StringFormat object. (Inherited from JsonContext.)
	WriteStyle	Writes a Style object.
	WriteSvg	Writes an SvgContent object.
	WriteTag	Writes the specified tag of the specified object.
	WriteTheme	Writes a Theme object.
	WriteThickness	Writes a Thickness object.

Protected Methods

	Name	Description
	Base64Decode	Decodes a string from Base64 representation. (Inherited from JsonContext.)
	Base64Encode	Encodes a string to Base64 representation. (Inherited from JsonContext.)
	ToCamelCase	Converts a string to Pascal case. (Inherited from JsonContext.)
	ToPascalCase	Converts a string to Camel case. (Inherited from JsonContext.)

See Also

JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext Fields

JsonPersistContext.LatestFormat Field

Specifies the latest file format revision number.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static int LatestFormat
```

Visual Basic Copy Code

```
Public Shared LatestFormat As Integer
```

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext Constructor

Overload List

Name	Description
JsonPersistContext (Diagram, Int32)	Initializes a new instance of the JsonPersistContext class.
JsonPersistContext (Diagram, Int32, Boolean)	Initializes a new instance of the JsonPersistContext class.

See Also

JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext Constructor (Diagram, Int32)

Initializes a new instance of the JsonPersistContext class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonPersistContext (
    Diagram diagram,
    int fileVersion
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram, _
    fileVersion As Integer _
)
```

Parameters

diagram
The Diagram that will be serialized or deserialized.

fileVersion
Specifies the current file format number.

See Also

JsonPersistContext Constructor Overload List
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext Constructor (Diagram, Int32, Boolean)

Initializes a new instance of the JsonPersistContext class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonPersistContext (
    Diagram diagram,
    int fileVersion,
    bool trackItems
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram, _
    fileVersion As Integer, _
    trackItems As Boolean _
)
```

Parameters

diagram
The Diagram that will be serialized or deserialized.

fileVersion
Specifies the current file format number.

trackItems
true to track shared item instances by their identifier, or false otherwise.

See Also

JsonPersistContext Constructor Overload List
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext Methods

JsonPersistContext.InitItem Method

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem InitItem (
    JsonObject obj
)
```

Visual Basic Copy Code

```
Public Function InitItem( _
    obj As JsonObject _
) As DiagramItem
```

Parameters

obj

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.LoadItem Method

Reads a diagram item from the specified JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem LoadItem (
    JsonObject obj
)
```

Visual Basic Copy Code

```
Public Function LoadItem( _
    obj As JsonObject _
) As DiagramItem
```

Parameters

obj

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadAnchorPattern Method

Reads an AnchorPattern object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern ReadAnchorPattern (
    JsonValue value
)
```

Visual Basic Copy Code

```
Public Function ReadAnchorPattern( _
    value As JsonValue _
) As AnchorPattern
```

Parameters

value

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadComposite Method

Reads the properties of a CompositeNode, which are marked with the AutoJson attribute.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ReadComposite (
    JsonObject obj,
    CompositeNode node
)
```

Visual Basic Copy Code

```
Public Sub ReadComposite( _
    obj As JsonObject, _
    node As CompositeNode _
)
```

Parameters

obj

node

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadEffect Method

Reads a NodeEffect object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeEffect ReadEffect (
    JsonValue value
)
```

Visual Basic Copy Code

```
Public Function ReadEffect( _
    value As JsonValue _
) As NodeEffect
```

Parameters

value

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadFont Method

Reads a Font object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Font ReadFont (
    JsonObject obj,
    Font defaultFont
)
```

Visual Basic Copy Code

```
Public Overrides Function ReadFont( _
    obj As JsonObject, _
    defaultFont As Font _
) As Font
```

Parameters

obj

defaultFont

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadItemId Method

Reads an item reference.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem ReadItemId (
    JsonValue value
)
```

Visual Basic Copy Code

```
Public Function ReadItemId( _
    value As JsonValue _
) As DiagramItem
```

Parameters

value

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadMap Method

Reads a MapContent object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MapContent ReadMap (
    JsonValue value
)
```

Visual Basic Copy Code

```
Public Function ReadMap( _
    value As JsonValue _
) As MapContent
```

Parameters

value

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadShape Method

Reads a Shape object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape ReadShape (
    JsonValue value
)
```

Visual Basic Copy Code

```
Public Function ReadShape( _
    value As JsonValue _
) As Shape
```

Parameters

value

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadStyle Method

Reads a Style object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ReadStyle (
    JsonValue value,
    Style style
)
```

Visual Basic Copy Code

```
Public Sub ReadStyle( _
    value As JsonValue, _
    style As Style _
)
```

Parameters

value

style

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadSvg Method

Reads a SVG objec.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgContent ReadSvg (
    JsonValue value
)
```

Visual Basic Copy Code

```
Public Function ReadSvg( _
    value As JsonValue _
) As SvgContent
```

Parameters

value

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadTag Method

Reads the specified tag for the specified object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object ReadTag (
    Object item,
    JsonValue tagValue,
    string propertyName
)
```

Visual Basic Copy Code

```
Public Function ReadTag( _
    item As Object, _
    tagValue As JsonValue, _
    propertyName As String _
) As Object
```

Parameters

item

tagValue

propertyName

Return Value

Remarks

Tags are read similarly to arbitrary objects except that the client of the diagram is given a chance to perform custom deserialization through event handling.

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadTheme Method

Reads a Theme object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Theme ReadTheme (
    JsonValue value
)
```

Visual Basic

```
Public Function ReadTheme( _
    value As JsonValue _
) As Theme
```

Parameters

value

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.ReadThickness Method

Reads a Thickness object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Thickness ReadThickness (
    JsonValue value
)
```

Visual Basic

```
Public Function ReadThickness( _
    value As JsonValue _
) As Thickness
```

Parameters

value

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.SaveItem Method

Saves the specified diagram item to a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public JsonObject SaveItem (
    DiagramItem item
)
```

Visual Basic

```
Public Function SaveItem( _
    item As DiagramItem _
) As JsonObject
```

Parameters

item

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteAnchorPattern Method

Writes an AnchorPattern object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public JsonValue WriteAnchorPattern (
    AnchorPattern anchorPattern
)
```

Visual Basic

```
Public Function WriteAnchorPattern( _
    anchorPattern As AnchorPattern _
) As JsonValue
```

Parameters

anchorPattern

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteComposite Method

Writes the properties of a CompositeNode, which are marked with the AutoJson attribute.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void WriteComposite (
    JsonObject obj,
    CompositeNode node
)
```

Visual Basic

```
Public Sub WriteComposite( _
    obj As JsonObject, _
    node As CompositeNode _
)
```

Parameters

obj

node

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteEffect Method

Writes a NodeEffect object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public JsonValue WriteEffect (
    NodeEffect effect
)
```

Visual Basic

```
Public Function WriteEffect( _
    effect As NodeEffect _
) As JsonValue
```

Parameters

effect

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteFont Method

Writes a Font object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override JsonObject WriteFont (
    Font font
)
```

Visual Basic

```
Public Overrides Function WriteFont( _
    font As Font _
) As JsonObject
```

Parameters

font

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteHandlesStyle Method

Writes a HandlesVisualStyle object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonValue WriteHandlesStyle (
    HandlesVisualStyle style
)
```

Visual Basic Copy Code

```
Public Function WriteHandlesStyle( _
    style As HandlesVisualStyle _
) As JsonValue
```

Parameters

style

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteItemId Method

Writes a reference to the specified item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonValue WriteItemId (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Function WriteItemId( _
    item As DiagramItem _
) As JsonValue
```

Parameters

item

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteItemType Method

Writes an item type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonValue WriteItemType (
    Type itemType
)
```

Visual Basic Copy Code

```
Public Function WriteItemType( _
    itemType As Type _
) As JsonValue
```

Parameters

itemType

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteMap Method

Writes a MapContent object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonValue WriteMap (
    MapContent map
)
```

Visual Basic Copy Code

```
Public Function WriteMap( _
    map As MapContent _
) As JsonValue
```

Parameters

map

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteShape Method

Writes a Shape object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonValue WriteShape (
    Shape shape
)
```

Visual Basic Copy Code

```
Public Function WriteShape( _
    shape As Shape _
) As JsonValue
```

Parameters

shape

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteStyle Method

Writes a Style object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonValue WriteStyle (
    Style style
)
```

Visual Basic Copy Code

```
Public Function WriteStyle( _
    style As Style _
) As JsonValue
```

Parameters

style

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteSvg Method

Writes an SvgContent object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonValue WriteSvg (
    SvgContent svg
)
```

Visual Basic Copy Code

```
Public Function WriteSvg( _
    svg As SvgContent _
) As JsonValue
```

Parameters

svg

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteTag Method

Writes the specified tag of the specified object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonValue WriteTag (
    Object item,
    Object tagValue,
    string propertyName
)
```

Visual Basic

```
Public Function WriteTag( _
    item As Object, _
    tagValue As Object, _
    propertyName As String _
) As JsonValue
```

Parameters

item

tagValue

propertyName

Return Value

Remarks

Tags are written similarly to arbitrary objects except that the client of the diagram is given a chance to perform custom serialization through event handling.

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteTheme Method

Writes a Theme object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public JsonObject WriteTheme (
    Theme theme
)
```

Visual Basic

```
Public Function WriteTheme( _
    theme As Theme _
) As JsonObject
```

Parameters

theme

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.WriteThickness Method

Writes a Thickness object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonObject WriteThickness (
    Thickness thickness
)
```

Visual Basic Copy Code

```
Public Function WriteThickness( _
    thickness As Thickness _
) As JsonObject
```

Parameters

thickness

Return Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext Properties

JsonPersistContext.Diagram Property

Gets the underlying diagram object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Diagram Diagram { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Diagram As Diagram
```

Property Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonPersistContext.FileVersion Property

Gets the format revision number for the file being currently serialized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int FileVersion { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property FileVersion As Integer
```

Property Value

See Also

JsonPersistContext Members
JsonPersistContext Class
MindFusion.Diagramming Namespace

JsonSerializeTagEventArgs Class

Contains the arguments passed to the JsonSerializeTag and JsonDeserializeTag event handlers.

Syntax

C# Copy Code

```
public class JsonSerializeTagEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class JsonSerializeTagEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.JsonSerializeTagEventArgs

See Also

JsonSerializeTagEventArgs Members
MindFusion.Diagramming Namespace

JsonSerializeTagEventArgs Members

Contains the arguments passed to the JsonSerializeTag and JsonDeserializeTag event handlers.

The following tables list the members exposed by the JsonSerializeTagEventArgs type.

Public Properties

	Name	Description
	Context	Gets a context object that can help you write and read various values.
	Handled	Gets or sets a value indicating whether the event is handled.
	Object	Gets the object being serialized/deserialized.
	PropertyName	Gets the name of the property being serialized.
	Representation	Gets or sets the JsonObject, representing the tag object being serialized / deserialized.
	Tag	Gets or sets the tag object being serialized/deserialized.

See Also

JsonSerializeTagEventArgs Properties

JsonSerializeTagEventArgs.Context Property

Gets a context object that can help you write and read various values.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonPersistContext Context { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Context As JsonPersistContext
```

Property Value

See Also

JsonSerializeTagEventArgs Members
JsonSerializeTagEventArgs Class
MindFusion.Diagramming Namespace

JsonSerializeTagEventArgs.Handled Property

Gets or sets a value indicating whether the event is handled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Handled { get; set; }
```

Visual Basic Copy Code

```
Public Property Handled As Boolean
```

Property Value

Remarks

Always set this flag to true if you perform custom tag serialization.

See Also

JsonSerializeTagEventArgs Members
JsonSerializeTagEventArgs Class
MindFusion.Diagramming Namespace

JsonSerializeTagEventArgs.Object Property

Gets the object being serialized/deserialized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Object { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Object As Object
```

Property Value

See Also

JsonSerializeTagEventArgs Members
JsonSerializeTagEventArgs Class
MindFusion.Diagramming Namespace

JsonSerializeTagEventArgs.PropertyName Property

Gets the name of the property being serialized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string PropertyName { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property PropertyName As String
```

Property Value

See Also

JsonSerializeTagEventArgs Members
JsonSerializeTagEventArgs Class
MindFusion.Diagramming Namespace

JsonSerializeTagEventArgs.Representation Property

Gets or sets the JsonObject, representing the tag object being serialized / deserialized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public JsonObject Representation { get; set; }
```

Visual Basic Copy Code

```
Public Property Representation As JsonObject
```

Property Value

See Also

JsonSerializeTagEventArgs Members
JsonSerializeTagEventArgs Class
MindFusion.Diagramming Namespace

JsonSerializeTagEventArgs.Tag Property

Gets or sets the tag object being serialized/deserialized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Tag { get; set; }
```

Visual Basic Copy Code

```
Public Property Tag As Object
```

Property Value

See Also

JsonSerializeTagEventArgs Members
JsonSerializeTagEventArgs Class
MindFusion.Diagramming Namespace

Key Enumeration

A copy of the **System.Windows.Forms.Key** enumeration used to propagate input information to manipulators and components.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum Key
```

Visual Basic Copy Code

```
Public Enum Key
```

Members

	Member name	Description
	Modifiers	The bitmask to extract modifiers from a key value.
	None	No key pressed.
	LButton	The left mouse button.
	RButton	The right mouse button.

	Cancel	The CANCEL key.
	MButton	The middle mouse button (three-button mouse).
	XButton1	The first x mouse button (five-button mouse).
	XButton2	The second x mouse button (five-button mouse).
	Back	The BACKSPACE key.
	Tab	The TAB key.
	LineFeed	The LINEFEED key.
	Clear	The CLEAR key.
	Enter	The ENTER key.
	Return	The RETURN key.
	ShiftKey	The SHIFT key.
	ControlKey	The CTRL key.
	Menu	The ALT key.
	Pause	The PAUSE key.
	Capital	The CAPS LOCK key.
	CapsLock	The CAPS LOCK key.
	KanaMode	The IME Kana mode key.
	HangulMode	The IME Hangul mode key.
	HanguelMode	The IME Hanguel mode key. (maintained for compatibility; use HangulMode)
	JunjaMode	The IME Junja mode key.
	FinalMode	The IME final mode key.
	KanjiMode	The IME Kanji mode key.
	HanjaMode	The IME Hanja mode key.
	Escape	The ESC key.
	IMEConvert	The IME convert key.
	IMENonconvert	The IME nonconvert key.
	IMEAceept	The IME accept key. Obsolete, use System.Windows.Forms.Keys.IMEAccept instead.
	IMEAccept	The IME accept key, replaces System.Windows.Forms.Keys.IMEAceept.
	IMEModeChange	The IME mode change key.
	Space	The SPACEBAR key.
	PageUp	The PAGE UP key.
	Prior	The PAGE UP key.
	PageDown	The PAGE DOWN key.
	Next	The PAGE DOWN key.
	End	The END key.
	Home	The HOME key.

	Left	The LEFT ARROW key.
	Up	The UP ARROW key.
	Right	The RIGHT ARROW key.
	Down	The DOWN ARROW key.
	Select	The SELECT key.
	Print	The PRINT key.
	Execute	The EXECUTE key.
	Snapshot	The PRINT SCREEN key.
	PrintScreen	The PRINT SCREEN key.
	Insert	The INS key.
	Delete	The DEL key.
	Help	The HELP key.
	D0	The 0 key.
	D1	The 1 key.
	D2	The 2 key.
	D3	The 3 key.
	D4	The 4 key.
	D5	The 5 key.
	D6	The 6 key.
	D7	The 7 key.
	D8	The 8 key.
	D9	The 9 key.
	A	The A key.
	B	The B key.
	C	The C key.
	D	The D key.
	E	The E key.
	F	The F key.
	G	The G key.
	H	The H key.
	I	The I key.
	J	The J key.
	K	The K key.
	L	The L key.
	M	The M key.
	N	The N key.
	O	The O key.

	P	The P key.
	Q	The Q key.
	R	The R key.
	S	The S key.
	T	The T key.
	U	The U key.
	V	The V key.
	W	The W key.
	X	The X key.
	Y	The Y key.
	Z	The Z key.
	LWin	The left Windows logo key (Microsoft Natural Keyboard).
	RWin	The right Windows logo key (Microsoft Natural Keyboard).
	Apps	The application key (Microsoft Natural Keyboard).
	Sleep	The computer sleep key.
	NumPad0	The 0 key on the numeric keypad.
	NumPad1	The 1 key on the numeric keypad.
	NumPad2	The 2 key on the numeric keypad.
	NumPad3	The 3 key on the numeric keypad.
	NumPad4	The 4 key on the numeric keypad.
	NumPad5	The 5 key on the numeric keypad.
	NumPad6	The 6 key on the numeric keypad.
	NumPad7	The 7 key on the numeric keypad.
	NumPad8	The 8 key on the numeric keypad.
	NumPad9	The 9 key on the numeric keypad.
	Multiply	The multiply key.
	Add	The add key.
	Separator	The separator key.
	Subtract	The subtract key.
	Decimal	The decimal key.
	Divide	The divide key.
	F1	The F1 key.
	F2	The F2 key.
	F3	The F3 key.
	F4	The F4 key.
	F5	The F5 key.
	F6	The F6 key.

	F7	The F7 key.
	F8	The F8 key.
	F9	The F9 key.
	F10	The F10 key.
	F11	The F11 key.
	F12	The F12 key.
	F13	The F13 key.
	F14	The F14 key.
	F15	The F15 key.
	F16	The F16 key.
	F17	The F17 key.
	F18	The F18 key.
	F19	The F19 key.
	F20	The F20 key.
	F21	The F21 key.
	F22	The F22 key.
	F23	The F23 key.
	F24	The F24 key.
	NumLock	The NUM LOCK key.
	Scroll	The SCROLL LOCK key.
	LShiftKey	The left SHIFT key.
	RShiftKey	The right SHIFT key.
	LControlKey	The left CTRL key.
	RControlKey	The right CTRL key.
	LMenu	The left ALT key.
	RMenu	The right ALT key.
	BrowserBack	The browser back key (Windows 2000 or later).
	BrowserForward	The browser forward key (Windows 2000 or later).
	BrowserRefresh	The browser refresh key (Windows 2000 or later).
	BrowserStop	The browser stop key (Windows 2000 or later).
	BrowserSearch	The browser search key (Windows 2000 or later).
	BrowserFavorites	The browser favorites key (Windows 2000 or later).
	BrowserHome	The browser home key (Windows 2000 or later).
	VolumeMute	The volume mute key (Windows 2000 or later).
	VolumeDown	The volume down key (Windows 2000 or later).
	VolumeUp	The volume up key (Windows 2000 or later).
	MediaNextTrack	The media next track key (Windows 2000 or later).

	MediaPreviousTrack	The media previous track key (Windows 2000 or later).
	MediaStop	The media Stop key (Windows 2000 or later).
	MediaPlayPause	The media play pause key (Windows 2000 or later).
	LaunchMail	The launch mail key (Windows 2000 or later).
	SelectMedia	The select media key (Windows 2000 or later).
	LaunchApplication1	The start application one key (Windows 2000 or later).
	LaunchApplication2	The start application two key (Windows 2000 or later).
	Oem1	The OEM 1 key.
	OemSemicolon	The OEM Semicolon key on a US standard keyboard (Windows 2000 or later).
	Oemplus	The OEM plus key on any country/region keyboard (Windows 2000 or later).
	Oemcomma	The OEM comma key on any country/region keyboard (Windows 2000 or later).
	OemMinus	The OEM minus key on any country/region keyboard (Windows 2000 or later).
	OemPeriod	The OEM period key on any country/region keyboard (Windows 2000 or later).
	Oem2	The OEM 2 key.
	OemQuestion	The OEM question mark key on a US standard keyboard (Windows 2000 or later).
	Oem3	The OEM 3 key.
	Oemtilde	The OEM tilde key on a US standard keyboard (Windows 2000 or later).
	Oem4	The OEM 4 key.
	OemOpenBrackets	The OEM open bracket key on a US standard keyboard (Windows 2000 or later).
	OemPipe	The OEM pipe key on a US standard keyboard (Windows 2000 or later).
	Oem5	The OEM 5 key.
	OemCloseBrackets	The OEM close bracket key on a US standard keyboard (Windows 2000 or later).
	Oem6	The OEM 6 key.
	OemQuotes	The OEM singled/double quote key on a US standard keyboard (Windows 2000 or later).
	Oem7	The OEM 7 key.
	Oem8	The OEM 8 key.
	Oem102	The OEM 102 key.
	OemBackslash	The OEM angle bracket or backslash key on the RT 102 key keyboard (Windows 2000 or later).
	ProcessKey	The PROCESS KEY key.
	Packet	Used to pass Unicode characters as if they were keystrokes. The Packet key value is the low word of a 32-bit virtual-key value used for non-keyboard input methods.
	Attn	The ATTN key.

	Crsel	The CRSEL key.
	Exsel	The EXSEL key.
	EraseEof	The ERASE EOF key.
	Play	The PLAY key.
	Zoom	The ZOOM key.
	NoName	A constant reserved for future use.
	Pa1	The PA1 key.
	OemClear	The CLEAR key.
	KeyCode	The bitmask to extract a key code from a key value.
	Shift	The SHIFT modifier key.
	Control	The CTRL modifier key.
	Alt	The ALT modifier key.

See Also

MindFusion.Diagramming Namespace

LabelPosition Enumeration

Specifies the position of labels in a **ListView** control.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum LabelPosition
```

Visual Basic Copy Code

```
Public Enum LabelPosition
```

Members

	Member name	Description
	Auto	Labels are positioned depending on the column count.
	Right	Labels are on the right side of the icon.
	Left	Labels are on the left side of the icon.
	Bottom	Labels are positioned below the icon.

See Also

MindFusion.Diagramming Namespace

Layer Class

Represents a layer in a Diagram object.

Syntax

C# Copy Code

```
public class Layer : INotifyPropertyChanged
```

Visual Basic Copy Code

```
Public Class Layer
    Implements INotifyPropertyChanged
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Layer

See Also

Layer Members
MindFusion.Diagramming Namespace

Layer Members

Represents a layer in a Diagram object.

The following tables list the members exposed by the Layer type.

Public Constructors

	Name	Description
	Layer	Initializes a new instance of the **Layer** class.

Public Properties

	Name	Description
	Locked	Gets or sets a value indicating whether the layer is locked.
	Title	Gets or sets a title associated with the layer.
	Visible	Gets or sets a value indicating whether the layer is visible.
	ZIndex	Gets or sets the z-index of the layer.

Protected Methods

	Name	Description
	OnPropertyChanged	Raises the PropertyChanged event.

Public Events

	Name	Description
	PropertyChanged	**INotifyPropertyChanged.PropertyChanged** implementation.

See Also

Layer Class
MindFusion.Diagramming Namespace

Layer Constructor

Initializes a new instance of the Layer class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Layer ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

Layer Class
MindFusion.Diagramming Namespace

Layer Methods

Layer.OnPropertyChanged Method

Raises the PropertyChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnPropertyChanged (
    PropertyChangedEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnPropertyChanged( _
    e As PropertyChangedEventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

Layer Members
Layer Class
MindFusion.Diagramming Namespace

Layer Properties

Layer.Locked Property

Gets or sets a value indicating whether the layer is locked.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Locked { get; set; }
```

Visual Basic Copy Code

```
Public Property Locked As Boolean
```

Property Value

true if the layer is locked; otherwise, **false**.

Remarks

Items associated with locked layers cannot be modified interactively, regardless of their own Locked property.

See Also

Layer Members
Layer Class
MindFusion.Diagramming Namespace

Layer.Title Property

Gets or sets a title associated with the layer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Title { get; set; }
```

Visual Basic Copy Code

```
Public Property Title As String
```

Property Value

A string value.

See Also

Layer Members
Layer Class
MindFusion.Diagramming Namespace

Layer.Visible Property

Gets or sets a value indicating whether the layer is visible.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Visible { get; set; }
```

Visual Basic

```
Public Property Visible As Boolean
```

Property Value

true if the layer is visible; otherwise, **false**.

Remarks

Items associated with hidden layers are not displayed, regardless of their own Visible property.

See Also

Layer Members
Layer Class
MindFusion.Diagramming Namespace

Layer.ZIndex Property

Gets or sets the z-index of the layer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public int ZIndex { get; set; }
```

Visual Basic

```
Public Property ZIndex As Integer
```

Property Value

An integer value.

See Also

Layer Members
Layer Class
MindFusion.Diagramming Namespace

Layer Events

Layer.PropertyChanged Event

INotifyPropertyChanged.PropertyChanged implementation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public event PropertyChangedEventHandler PropertyChanged
```

Visual Basic

```
Public Event PropertyChanged As PropertyChangedEventHandler
```

Event Data

PropertyChanged event handlers receive an argument of type **PropertyChangedEventArgs**.

See Also

Layer Members
Layer Class
MindFusion.Diagramming Namespace

LayerEventArgs Class

Contains the arguments passed to the LayerVisibilityChanged event handler.

Syntax

C# Copy Code

```
public class LayerEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class LayerEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.LayerEventArgs

See Also

LayerEventArgs Members
MindFusion.Diagramming Namespace

LayerEventArgs Members

Contains the arguments passed to the LayerVisibilityChanged event handler.

The following tables list the members exposed by the LayerEventArgs type.

Public Constructors

	Name	Description
	LayerEventArgs	Initializes a new instance of the LayerEventArgs class.

Public Properties

	Name	Description
	Index	Gets or sets the layer index related to the current event.
	Layer	Gets or sets the Layer related to the current event.

See Also

LayerEventArgs Class
MindFusion.Diagramming Namespace

LayerEventArgs Constructor

Initializes a new instance of the LayerEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LayerEventArgs (
    Layer layer,
    int index
)
```

Visual Basic Copy Code

```
Public New ( _
    layer As Layer, _
    index As Integer _
)
```

Parameters

layer

index

See Also

LayerEventArgs Class
MindFusion.Diagramming Namespace

LayerEventArgs Properties

LayerEventArgs.Index Property

Gets or sets the layer index related to the current event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Index { get; set; }
```

Visual Basic Copy Code

```
Public Property Index As Integer
```

Property Value

See Also

LayerEventArgs Members
LayerEventArgs Class
MindFusion.Diagramming Namespace

LayerEventArgs.Layer Property

Gets or sets the Layer related to the current event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Layer Layer { get; set; }
```

Visual Basic Copy Code

```
Public Property Layer As Layer
```

Property Value

See Also

LayerEventArgs Members
LayerEventArgs Class
MindFusion.Diagramming Namespace

LineTemplate Class

Represents a line segment in a shape definition.

Syntax

C# Copy Code

```
public class LineTemplate : ElementTemplate
```

Visual Basic Copy Code

```
Public Class LineTemplate
    Inherits ElementTemplate
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ElementTemplate
    MindFusion.Diagramming.LineTemplate

See Also

LineTemplate Members
MindFusion.Diagramming Namespace

LineTemplate Members

Represents a line segment in a shape definition.

The following tables list the members exposed by the LineTemplate type.

Public Constructors

	Name	Description
	LineTemplate	Overloaded. Initializes a new instance of the **LineTemplate** class.

Public Properties

	Name	Description
	Color	Gets or sets the outline color of this element. (Inherited from ElementTemplate.)

	Coordinates	Gets a double array containing the coordinates of the end points of this line segment.
	DashStyle	Gets or sets the dash style of the outline of this element. (Inherited from ElementTemplate.)
	Width	Gets or sets the outline width of this element. (Inherited from ElementTemplate.)

Public Methods

	Name	Description
	Clone	Create an exact copy of this **LineTemplate** element.
	SameAs	ElementTemplate.SameAs override.

Protected Methods

	Name	Description
	GetElementPen	Gets a pen used to render the element. If no pen properties are explicitly specified for this element, returns the pen passed as an argument. (Inherited from ElementTemplate.)

See Also

LineTemplate Class
MindFusion.Diagramming Namespace

LineTemplate Constructor

Initializes a new instance of the LineTemplate class.

Overload List

Name	Description
LineTemplate (Double, Double, Double, Double)	Initializes a new instance of the **LineTemplate** class with the specified start and end point.
LineTemplate (Double, Double, Double, Double, Color, DashStyle, Double)	Initializes a new instance of the **LineTemplate** class with the specified start and end points, and the specified color and style.

See Also

LineTemplate Class
MindFusion.Diagramming Namespace

LineTemplate Constructor (Double, Double, Double, Double)

Initializes a new instance of the LineTemplate class with the specified start and end point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LineTemplate (
    double x1,
    double y1,
    double x2,
```

```
    double y2
)
```

Visual Basic Copy Code

```
Public New ( _
    x1 As Double, _
    y1 As Double, _
    x2 As Double, _
    y2 As Double _
)
```

Parameters

x1
The x-coordinate of the first point, expressed as percentage of a shape node's bounding rectangle.

y1
The y-coordinate of the first point, expressed as percentage of a shape node's bounding rectangle.

x2
The x-coordinate of the second point, expressed as percentage of a shape node's bounding rectangle.

y2
The y-coordinate of the second point, expressed as percentage of a shape node's bounding rectangle.

Remarks

A line template is defined by two points whose coordinates are passed as arguments.

See Also

LineTemplate Constructor Overload List
LineTemplate Class
MindFusion.Diagramming Namespace

LineTemplate Constructor (Double, Double, Double, Double, Color, DashStyle, Double)

Initializes a new instance of the LineTemplate class with the specified start and end points, and the specified color and style.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LineTemplate (
    double x1,
    double y1,
    double x2,
    double y2,
    Color color,
    DashStyle dashStyle,
    double width
)
```

Visual Basic Copy Code

```
Public New ( _
    x1 As Double, _
    y1 As Double, _
    x2 As Double, _
    y2 As Double, _
    color As Color, _
    dashStyle As DashStyle, _
    width As Double _
)
```

Parameters

x1
The x-coordinate of the first point, expressed as percentage of a shape node's bounding rectangle.

y1
The y-coordinate of the first point, expressed as percentage of a shape node's bounding rectangle.

x2
The x-coordinate of the second point, expressed as percentage of a shape node's bounding rectangle.

y2
The y-coordinate of the second point, expressed as percentage of a shape node's bounding rectangle.

color
The color of this segment.

dashStyle
The dash style of this segment.

width
The width of this segment.

Remarks

A line template is defined by two points whose coordinates are passed as arguments.

See Also

LineTemplate Constructor Overload List
LineTemplate Class
MindFusion.Diagramming Namespace

LineTemplate Methods

LineTemplate.Clone Method

Create an exact copy of this LineTemplate element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Object Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Object
```

Return Value

An instance of the LineTemplate class.

Remarks

An instance of the LineTemplate class.

See Also

LineTemplate Members
LineTemplate Class
MindFusion.Diagramming Namespace

LineTemplate.SameAs Method

ElementTemplate.SameAs override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool SameAs (
    ElementTemplate obj
)
```

Visual Basic Copy Code

```
Public Overrides Function SameAs( _
    obj As ElementTemplate _
) As Boolean
```

Parameters

obj
The ElementTemplate to compare to this template.

Return Value

true if the specified object is a LineTemplate closely approximating this template; otherwise, **false**.

See Also

LineTemplate Members
LineTemplate Class
MindFusion.Diagramming Namespace

LineTemplate Properties

LineTemplate.Coordinates Property

Gets a double array containing the coordinates of the end points of this line segment.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double[] Coordinates { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Coordinates As Double()
```

Property Value

An array of **double** values containing the coordinates of the points composing this element.

See Also

LineTemplate Members
LineTemplate Class
MindFusion.Diagramming Namespace

LinkAdjustmentHandle Class

Represents an adjustment handle used to modify the position of a link's control point.

Syntax

C# Copy Code

```
public class LinkAdjustmentHandle : AdjustmentHandle
```

Visual Basic Copy Code

```
Public Class LinkAdjustmentHandle
    Inherits AdjustmentHandle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.AdjustmentHandle
    MindFusion.Diagramming.LinkAdjustmentHandle

See Also

LinkAdjustmentHandle Members
MindFusion.Diagramming Namespace

LinkAdjustmentHandle Members

Represents an adjustment handle used to modify the position of a link's control point.

The following tables list the members exposed by the LinkAdjustmentHandle type.

Public Constructors

	Name	Description
	LinkAdjustmentHandle	Initializes a new instance of the **LinkAdjustmentHandle** class.

Public Properties

	Name	Description
	PointIndex	Gets or sets the index of a link's control point that will be adjusted by this handle.

Public Methods

	Name	Description
	IsMoveHandle	Returns a value indicating whether this handle is used to move items. (Inherited from AdjustmentHandle.)
	IsResizeHandle	Gets whether this adjustment handle resizes the item. (Inherited from AdjustmentHandle.)
	IsRotationHandle	Gets whether this adjustment handle changes item's rotation angle. (Inherited from AdjustmentHandle.)

See Also

LinkAdjustmentHandle Class
MindFusion.Diagramming Namespace

LinkAdjustmentHandle Constructor

Initializes a new instance of the LinkAdjustmentHandle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkAdjustmentHandle (
    int pointIndex
)
```

Visual Basic Copy Code

```
Public New ( _
    pointIndex As Integer _
)
```

Parameters

pointIndex
An integer value specifying the index of a link's control point that will be adjusted by this handle.

Remarks

When the user moves this handle, the respective element in link's ControlPoints collection is set to the handle's position.

See Also

LinkAdjustmentHandle Class
MindFusion.Diagramming Namespace

LinkAdjustmentHandle Properties

LinkAdjustmentHandle.PointIndex Property

Gets or sets the index of a link's control point that will be adjusted by this handle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int PointIndex { get; set; }
```

Visual Basic Copy Code

```
Public Property PointIndex As Integer
```

Property Value

An integer index specifying which element of link's ControlPoints collection is updated by this handle.

See Also

LinkAdjustmentHandle Members
LinkAdjustmentHandle Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs Class

Contains the arguments passed to the ValidateLinkCrossing event handler.

Syntax

C# Copy Code

```
public class LinkCrossingEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class LinkCrossingEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.LinkCrossingEventArgs

See Also

LinkCrossingEventArgs Members
MindFusion.Diagramming Namespace

LinkCrossingEventArgs Members

Contains the arguments passed to the ValidateLinkCrossing event handler.

The following tables list the members exposed by the LinkCrossingEventArgs type.

Public Constructors

	Name	Description
	LinkCrossingEventArgs	

Public Properties

	Name	Description
	BottomLink	The link at lower Z position.
	BottomSegmentIndex	Index of bottom link's segment that contains the intersection point.
	Cancel	Specifies whether crossing decoration should be displayed.
	IntersectionPoint	Intersection point that will show crossing decoration if the event is validated.
	TopLink	The link at higher Z position.
	TopSegmentIndex	Index of top link's segment that contains the intersection point.

See Also

LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs Constructor

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkCrossingEventArgs ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs Properties

LinkCrossingEventArgs.BottomLink Property

The link at lower Z position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink BottomLink { get; set; }
```

Visual Basic Copy Code

```
Public Property BottomLink As DiagramLink
```

Property Value

A DiagramLink reference.

See Also

LinkCrossingEventArgs Members
LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs.BottomSegmentIndex Property

Index of bottom link's segment that contains the intersection point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int BottomSegmentIndex { get; set; }
```

Visual Basic Copy Code

```
Public Property BottomSegmentIndex As Integer
```

Property Value

An integer index of link's segment containing IntersectionPoint.

See Also

LinkCrossingEventArgs Members
LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs.Cancel Property

Specifies whether crossing decoration should be displayed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Cancel { get; set; }
```

Visual Basic Copy Code

```
Public Property Cancel As Boolean
```

Property Value

true to hide the crossing decoration, or **false** to display it.

See Also

LinkCrossingEventArgs Members
LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs.IntersectionPoint Property

Intersection point that will show crossing decoration if the event is validated.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point IntersectionPoint { get; set; }
```

Visual Basic Copy Code

```
Public Property IntersectionPoint As Point
```

Property Value

A **Microsoft.Maui.Graphics.Point** containing intersection coordinates.

See Also

LinkCrossingEventArgs Members
LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs.TopLink Property

The link at higher Z position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink TopLink { get; set; }
```

Visual Basic

```
Public Property TopLink As DiagramLink
```

Property Value

A DiagramLink reference.

See Also

LinkCrossingEventArgs Members
LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossingEventArgs.TopSegmentIndex Property

Index of top link's segment that contains the intersection point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public int TopSegmentIndex { get; set; }
```

Visual Basic

```
Public Property TopSegmentIndex As Integer
```

Property Value

An integer index of link's segment containing IntersectionPoint.

See Also

LinkCrossingEventArgs Members
LinkCrossingEventArgs Class
MindFusion.Diagramming Namespace

LinkCrossings Enumeration

Defines values that indicate how to render link intersection points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public enum LinkCrossings
```

Visual Basic

```
Public Enum LinkCrossings
```

Members

	Member name	Description
	Straight	No special indication for link intersection points.
	Arcs	An arc is drawn at the intersection point of two links. It connects the segments of the link at higher Z position and goes over the link at lower Z.

	Cut	The intersection segments of links at lower Z position are cut by links at higher Z position.

Remarks

Values from this enumeration can be assigned to LinkCrossings.

See Also

MindFusion.Diagramming Namespace

LinkEventArgs Class

Contains the arguments passed to handlers of link related events, such as LinkCreated and LinkDeleted.

Syntax

C# Copy Code

```
public class LinkEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class LinkEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.LinkEventArgs
      MindFusion.Diagramming.LinkShapeEventArgs
      MindFusion.Diagramming.LinkValidationEventArgs
      MindFusion.Diagramming.LinkValidationEventArgs

See Also

LinkEventArgs Members
MindFusion.Diagramming Namespace

LinkEventArgs Members

Contains the arguments passed to handlers of link related events, such as LinkCreated and LinkDeleted.

The following tables list the members exposed by the LinkEventArgs type.

Public Constructors

	Name	Description
	LinkEventArgs	Overloaded. Initializes a new instance of the **LinkEventArgs** class.

Public Properties

	Name	Description
	AdjustmentHandle	Gets or sets the adjustment handle being moved by the user.
	Label	Gets or sets a LinkLabel related to this event.
	Link	Gets the link for which the event raised.
	MouseButton	Gets which mouse button has been pressed.
	MousePosition	Gets or sets the current mouse pointer position.

See Also

LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs Constructor

Initializes a new instance of the LinkEventArgs class.

Overload List

Name	Description
LinkEventArgs (DiagramLink)	Initializes a new instance of the **LinkEventArgs** class.
LinkEventArgs (DiagramLink, MouseButton, Point)	Initializes a new instance of the **LinkEventArgs** class.
LinkEventArgs (DiagramLink, Point, Int32)	Initializes a new instance of the **LinkEventArgs** class.

See Also

LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs Constructor (DiagramLink)

Initializes a new instance of the LinkEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkEventArgs (
    DiagramLink link
)
```

Visual Basic Copy Code

```
Public New ( _
    link As DiagramLink _
)
```

Parameters

link
The DiagramLink related to the event.

See Also

LinkEventArgs Constructor Overload List
LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs Constructor (DiagramLink, MouseButton, Point)

Initializes a new instance of the LinkEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkEventArgs (
    DiagramLink link,
    MouseButton mouseButton,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public New ( _
    link As DiagramLink, _
    mouseButton As MouseButton, _
    mousePosition As Point _
)
```

Parameters

link
The DiagramLink related to the event.

mouseButton
The mouse button related to the event.

mousePosition
The position of the mouse cursor, in document coordinates.

See Also

LinkEventArgs Constructor Overload List
LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs Constructor (DiagramLink, Point, Int32)

Initializes a new instance of the LinkEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkEventArgs (
    DiagramLink link,
    Point mousePosition,
    int adjustmentHandle
)
```

Visual Basic Copy Code

```
Public New ( _
    link As DiagramLink, _
    mousePosition As Point, _
    adjustmentHandle As Integer _
)
```

Parameters

link
The DiagramLink related to the event.

mousePosition
The position of the mouse cursor, in document coordinates.

adjustmentHandle
The zero-based index of the control point related to the event.

See Also

LinkEventArgs Constructor Overload List
LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs Properties

LinkEventArgs.AdjustmentHandle Property

Gets or sets the adjustment handle being moved by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int AdjustmentHandle { get; set; }
```

Visual Basic Copy Code

```
Public Property AdjustmentHandle As Integer
```

Property Value

An integer that represents the index of the link's control point being moved.

See Also

LinkEventArgs Members
LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs.Label Property

Gets or sets a LinkLabel related to this event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkLabel Label { get; set; }
```

Visual Basic Copy Code

```
Public Property Label As LinkLabel
```

Property Value

See Also

LinkEventArgs Members
LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs.Link Property

Gets the link for which the event raised.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink Link { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Link As DiagramLink
```

Property Value

A DiagramLink instance.

See Also

LinkEventArgs Members
LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs.MouseButton Property

Gets which mouse button has been pressed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MouseButton MouseButton { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MouseButton As MouseButton
```

Property Value

A member of the MouseButton enumeration specifying the currently pressed mouse button.

See Also

LinkEventArgs Members
LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkEventArgs.MousePosition Property

Gets or sets the current mouse pointer position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point MousePosition { get; set; }
```

Visual Basic

```
Public Property MousePosition As Point
```

Property Value

A **Microsoft.Maui.Graphics.Point** specifying the current mouse position.

See Also

LinkEventArgs Members
LinkEventArgs Class
MindFusion.Diagramming Namespace

LinkLabel Class

Represents a supplementary label displayed by a DiagramLink. New labels can be added to a link by calling the AddLabel method.

Syntax

C#

```
public class LinkLabel : ItemLabel
```

Visual Basic

```
Public Class LinkLabel
    Inherits ItemLabel
```

Remarks

The text displayed by a **LinkLabel** is set via the Text property, and its attributes are specified via Font. The label's foreground and background can be set via TextBrush and Brush properties.

The position of a label is determined by where it is pinned along the link's geometry, as set using RelativeTo and Index properties, and offset from the pin point specified by HorizontalOffset and VerticalOffset properties. The SetControlPointPosition, SetSegmentPosition and SetLinkLengthPosition methods let you specify the values as arguments. Instead of using fixed offsets, enable the AutoArrange property to let links automatically choose positions of their labels that won't intersect nodes or other labels.

You can let users arrange labels manually by setting the **Behavior** property of **DiagramView** to MoveLabels.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ItemLabel
    MindFusion.Diagramming.LinkLabel

See Also

LinkLabel Members
MindFusion.Diagramming Namespace

LinkLabel Members

Represents a supplementary label displayed by a DiagramLink. New labels can be added to a link by calling the AddLabel method.

The following tables list the members exposed by the LinkLabel type.

Public Constructors

	Name	Description
	LinkLabel	Overloaded. Initializes a new instance of the **LinkLabel** class.

Public Properties

	Name	Description
	AutoArrange	Gets or sets a value indicating whether the label can be auto arranged.
	Brush	Gets or sets the brush used to draw the label's background.
	ContentPadding	Gets or sets the distance between image and text when both are specified. (Inherited from ItemLabel.)
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled. (Inherited from ItemLabel.)
	EnableWrap	Gets or sets a value indicating whether the label's text can be wrapped.
	Font	Gets or sets the label's font.
	HorizontalAlign	Gets or sets the label's horizontal alignment.
	HorizontalOffset	Gets or sets the label's horizontal offset.
	Image	Gets or sets the image displayed by this label. (Inherited from ItemLabel.)
	ImageAlign	Gets or sets the alignment of label's image relative to text. (Inherited from ItemLabel.)
	ImageSize	Gets or sets a size that overrides the native image size. (Inherited from ItemLabel.)
	ImageUrl	Gets or sets URL of the image to display inside this label. (Inherited from ItemLabel.)
	Index	Gets or sets the index of the segment or control point this label is relative to.
	Item	Gets or sets the DiagramItem that displays this label. (Inherited from ItemLabel.)
	LengthFactor	Gets or sets label's position along link's length when the RelativeTo property is set to LinkLength.
	Link	Gets or sets the **DiagramLink** that displays this label.
	Locked	Gets or sets a value indicating whether users are allowed to edit this label. (Inherited from ItemLabel.)
	Margin	Gets or sets the label's margin size.
	MaxWidth	Gets or sets the maximum width of the label.
	RelativeTo	Gets or sets the label's relative positioning.
	Text	Gets or sets the label's text.
	TextBrush	Gets or sets the brush used to draw the label's text.
	VerticalAlign	Gets or sets the label's vertical alignment.
	VerticalOffset	Gets or sets the label's vertical offset.

Public Methods

	Name	Description
	Cancel	Called when drag operation is canceled. (Inherited from ItemLabel.)
	Clone	Creates a clone of this label.
	Draw	Draws the link label on the specified IGraphics surface.
	GetEditRect	InplaceEditable.GetEditRect implementation.
	GetTextToEdit	InplaceEditable.GetTextToEdit implementation.

	LoadFromXml	Loads the label from the specified XML element.
	OnMouseDown	Called when the left mouse button is pressed while the pointer is over this label. (Inherited from ItemLabel.)
	OnMouseMove	Called when the mouse is moved while this label tracks the mouse events. (Inherited from ItemLabel.)
	OnMouseUp	Called when the left mouse button is released while this label tracks the mouse events. (Inherited from ItemLabel.)
	SaveToXml	Saves the label into the specified XML element.
	SetControlPointPosition	Positions this label relatively to the specified control point.
	SetEditedText	InplaceEditable.SetEditedText implementation.
	SetLinkLengthPosition	Positions this label relatively to the link length.
	SetMouseCursor	Indicates what cursor to display for interactive labels. (Inherited from ItemLabel.)
	SetSegmentPosition	Positions this label relatively to the specified link segment.
	UpdateValue	Called while drag operation is in progress. (Inherited from ItemLabel.)

Protected Methods

	Name	Description
	LoadFromJson	Loads the label from a JSON object.
	PointAlongLength	Returns a point lying on a polyline that is positioned at a relative position along the polyline's length.
	SaveToJson	Saves the label as a JSON object.

See Also

LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel Constructor

Initializes a new instance of the LinkLabel class.

Overload List

Name	Description
LinkLabel ()	Initializes a new instance of the **LinkLabel** class.
LinkLabel (DiagramLink)	Initializes a new instance of the **LinkLabel** class.
LinkLabel (DiagramLink, String)	Initializes a new instance of the **LinkLabel** class.

See Also

LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel Constructor ()

Initializes a new instance of the LinkLabel class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkLabel ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

LinkLabel Constructor Overload List
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel Constructor (DiagramLink)

Initializes a new instance of the LinkLabel class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkLabel (
    DiagramLink link
)
```

Visual Basic Copy Code

```
Public New ( _
    link As DiagramLink _
)
```

Parameters

link
The DiagramLink that displays the label.

See Also

LinkLabel Constructor Overload List
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel Constructor (DiagramLink, String)

Initializes a new instance of the LinkLabel class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkLabel (
    DiagramLink link,
    string text
)
```

Visual Basic

```
Public New ( _
    link As DiagramLink, _
    text As String _
)
```

Parameters

link

The DiagramLink that displays the label.

text

The label's text.

See Also

LinkLabel Constructor Overload List
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel Methods

LinkLabel.Clone Method

Creates a clone of this label.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public LinkLabel Clone ()
```

Visual Basic

```
Public Function Clone () As LinkLabel
```

Return Value

A new LinkLabel instance that is an exact copy of this label.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.Draw Method

Draws the link label on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public virtual void Draw (
    IGraphics graphics
)
```

Visual Basic

```
Public Overridable Sub Draw( _
    graphics As IGraphics _
)
```

Parameters

graphics
An IGraphics object where the link label should be rendered.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.GetEditRect Method

InplaceEditable.GetEditRect implementation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Rect GetEditRect (
    DiagramItem item,
    Point mousePosition
)
```

Visual Basic

```
Public Function GetEditRect( _
    item As DiagramItem, _
    mousePosition As Point _
) As Rect
```

Parameters

item
The item being edited.

mousePosition
A **Microsoft.Maui.Graphics.Point** specifying the mouse position.

Return Value

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.GetTextToEdit Method

InplaceEditable.GetTextToEdit implementation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public string GetTextToEdit ()
```

Visual Basic

```
Public Function GetTextToEdit () As String
```

Return Value

A string value.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.LoadFromJson Method

Loads the label from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject containing the label's data.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.LoadFromXml Method

Loads the label from the specified XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic

```
Public Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

A **XmlElement** instance.

context

A XmlPersistContext instance.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.PointAlongLength Method

Returns a point lying on a polyline that is positioned at a relative position along the polyline's length.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected Point PointAlongLength (
    double lengthFactor,
    IList<Point> segments
)
```

Visual Basic Copy Code

```
Protected Function PointAlongLength( _
    lengthFactor As Double, _
    segments As IList(Of Point) _
) As Point
```

Parameters

lengthFactor

A value from 0 to 1 specifying a portion of link's length.

segments

A list of points specifying the polyline segments.

Return Value

A **Microsoft.Maui.Graphics.Point** instance.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.SaveToJson Method

Saves the label as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject instance to contain the serialized data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.SaveToXml Method

Saves the label into the specified XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic

```
Public Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
A **XmlElement** instance.

context
A XmlPersistContext instance.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.SetControlPointPosition Method

Positions this label relatively to the specified control point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetControlPointPosition (
    int controlPoint,
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Public Sub SetControlPointPosition( _
    controlPoint As Integer, _
    dx As Double, _
    dy As Double _
)
```

Parameters

controlPoint
The control point index.

dx
Horizontal offset from the point.

dy
Vertical offset from the point.

Remarks

This method sets the specified control point as label's pin point and assigns specified offsets to HorizontalOffset and VerticalOffset properties.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.SetEditedText Method

InplaceEditable.SetEditedText implementation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetEditedText (
    string newText
)
```

Visual Basic Copy Code

```
Public Sub SetEditedText( _
    newText As String _
)
```

Parameters

newText
The new text of the in-place edited object.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.SetLinkLengthPosition Method

Positions this label relatively to the link length.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetLinkLengthPosition (
    double lengthFactor
)
```

Visual Basic Copy Code

```
Public Sub SetLinkLengthPosition( _
    lengthFactor As Double _
)
```

Parameters

lengthFactor
A value between 0 and 1 specifying the position of the label relatively to the link's length.

Remarks

This method sets the point at specified distance along link's geometry as pin point and assigns specified offsets to HorizontalOffset and VerticalOffset properties.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.SetSegmentPosition Method

Positions this label relatively to the specified link segment.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetSegmentPosition (
    int segment,
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Public Sub SetSegmentPosition( _
    segment As Integer, _
    dx As Double, _
    dy As Double _
)
```

Parameters

segment
The segment index.

dx
Horizontal offset from the segment center.

dy
Vertical offset from the segment center.

Remarks

This method sets the middle of specified segment as label's pin point and assigns specified offsets to HorizontalOffset and VerticalOffset properties.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel Properties

LinkLabel.AutoArrange Property

Gets or sets a value indicating whether the label can be auto arranged.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AutoArrange { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoArrange As Boolean
```

Property Value

true to automatically place the label; otherwise, **false**.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.Brush Property

Gets or sets the brush used to draw the label's background.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush Brush { get; set; }
```

Visual Basic Copy Code

```
Public Property Brush As Brush
```

Property Value

A Brush used to paint the label's background. The default value is **null**.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.EnableWrap Property

Gets or sets a value indicating whether the label's text can be wrapped.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableWrap { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableWrap As Boolean
```

Property Value

true to enable text wrapping; otherwise, **false**.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.Font Property

Gets or sets the label's font.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Font Font { get; set; }
```

Visual Basic Copy Code

```
Public Property Font As Font
```

Property Value

A Font object.

Remarks

This property can be **null**, in which case the label will use the Font of its parent link.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.HorizontalAlign Property

Gets or sets the label's horizontal alignment.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringAlignment HorizontalAlign { get; set; }
```

Visual Basic Copy Code

```
Public Property HorizontalAlign As StringAlignment
```

Property Value

A member of the **StringAlignment** enumeration. The default is **Center**.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.HorizontalOffset Property

Gets or sets the label's horizontal offset.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public float HorizontalOffset { get; set; }
```

Visual Basic Copy Code

```
Public Property HorizontalOffset As Single
```

Property Value

A float value.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.Index Property

Gets or sets the index of the segment or control point this label is relative to.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Index { get; set; }
```

Visual Basic Copy Code

```
Public Property Index As Integer
```

Property Value

An integer value.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.LengthFactor Property

Gets or sets label's position along link's length when the RelativeTo property is set to LinkLength.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double LengthFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property LengthFactor As Double
```

Property Value

A **double** value in 0-1 range. The default is **0.5**.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace
SetLinkLengthPosition Method

LinkLabel.Link Property

Gets or sets the DiagramLink that displays this label.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink Link { get; set; }
```

Visual Basic Copy Code

```
Public Property Link As DiagramLink
```

Property Value

A DiagramLink object.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.Margin Property

Gets or sets the label's margin size.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Thickness Margin { get; set; }
```

Visual Basic Copy Code

```
Public Property Margin As Thickness
```

Property Value

A Thickness instance specifying the size of text margins.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.MaxWidth Property

Gets or sets the maximum width of the label.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public float MaxWidth { get; set; }
```

Visual Basic Copy Code

```
Public Property MaxWidth As Single
```

Property Value

A float value.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.RelativeTo Property

Gets or sets the label's relative positioning.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RelativeToLink RelativeTo { get; set; }
```

Visual Basic Copy Code

```
Public Property RelativeTo As RelativeToLink
```

Property Value

A member of the RelativeToLink enumeration. The default is **LinkLength**.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.Text Property

Gets or sets the label's text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Text { get; set; }
```

Visual Basic Copy Code

```
Public Property Text As String
```

Property Value

A string value.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.TextBrush Property

Gets or sets the brush used to draw the label's text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush TextBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property TextBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.VerticalAlign Property

Gets or sets the label's vertical alignment.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringAlignment VerticalAlign { get; set; }
```

Visual Basic Copy Code

```
Public Property VerticalAlign As StringAlignment
```

Property Value

A member of the **StringAlignment** enumeration. The default is **Far**.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkLabel.VerticalOffset Property

Gets or sets the label's vertical offset.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public float VerticalOffset { get; set; }
```

Visual Basic Copy Code

```
Public Property VerticalOffset As Single
```

Property Value

A float value.

See Also

LinkLabel Members
LinkLabel Class
MindFusion.Diagramming Namespace

LinkShape Enumeration

Specifies available styles for the segments of links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum LinkShape
```

Visual Basic Copy Code

```
Public Enum LinkShape
```

Members

	Member name	Description
	Bezier	The segments of a link are Bézier curves.
	Polyline	The segments of the link are straight lines.
	Cascading	The segments of a link are horizontal and vertical lines, each one orthogonal to its adjacent segments.
	Spline	The segments of a link are curves forming an interpolating spline that passes through all control points.

Remarks

Values from this enumeration can be assigned to the Shape property of links or to the LinkShape property of the diagram.

See Also

MindFusion.Diagramming Namespace

LinkShapeEventArgs Class

Contains the arguments passed to the SetSelfLoopShape event handler.

Syntax

C# Copy Code

```
public class LinkShapeEventArgs : LinkEventArgs
```

Visual Basic Copy Code

```
Public Class LinkShapeEventArgs
    Inherits LinkEventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.LinkEventArgs
      MindFusion.Diagramming.LinkShapeEventArgs

See Also

LinkShapeEventArgs Members
MindFusion.Diagramming Namespace

LinkShapeEventArgs Members

Contains the arguments passed to the SetSelfLoopShape event handler.

The following tables list the members exposed by the LinkShapeEventArgs type.

Public Constructors

	Name	Description
	LinkShapeEventArgs	Initializes a new instance of the **LinkShapeEventArgs** class.

Public Properties

	Name	Description
	AdjustmentHandle	Gets or sets the adjustment handle being moved by the user. (Inherited from LinkEventArgs.)
	Handled	Indicates whether the event handler has set a custom loop shape.
	Label	Gets or sets a LinkLabel related to this event. (Inherited from LinkEventArgs.)
	Link	Gets the link for which the event raised. (Inherited from LinkEventArgs.)
	MouseButton	Gets which mouse button has been pressed. (Inherited from LinkEventArgs.)
	MousePosition	Gets or sets the current mouse pointer position. (Inherited from LinkEventArgs.)

See Also

LinkShapeEventArgs Class
MindFusion.Diagramming Namespace

LinkShapeEventArgs Constructor

Initializes a new instance of the LinkShapeEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkShapeEventArgs (
    DiagramLink link
)
```

Visual Basic Copy Code

```
Public New ( _
    link As DiagramLink _
)
```

Parameters

link
The DiagramLink that becomes a self-loop.

See Also

LinkShapeEventArgs Class
MindFusion.Diagramming Namespace

LinkShapeEventArgs Properties

LinkShapeEventArgs.Handled Property

Indicates whether the event handler has set a custom loop shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Handled { get; set; }
```

Visual Basic Copy Code

```
Public Property Handled As Boolean
```

Property Value

true if a custom shape is specified and the control should not set the standard loop shape; otherwise, **false**. The default value is **false**.

See Also

LinkShapeEventArgs Members
LinkShapeEventArgs Class
MindFusion.Diagramming Namespace

LinkTextStyle Enumeration

Specifies the position and alignment of links' labels.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum LinkTextStyle
```

Visual Basic Copy Code

```
Public Enum LinkTextStyle
```

Members

	Member name	Description
	Center	The text is displayed horizontally above the middle link segment or control point, depending on whether there are an odd or even number of segments.
	Rotate	The text is displayed rotated at the same angle as the segment where the text is placed. A long enough segment from the middle link segments is chosen for that purpose.
	OverLongestSegment	The text follows the longest link segment and is displayed centered at the segment's middle point.
	Follow	The label text starts from the first point of the link and follows the path defined by the link segments. This style is useful for displaying long text.
	MiddleSegment	The text is displayed horizontally above the middle link segment. If there are an even number of segments, the label is drawn over the longer from the two segments adjacent to the middle control point.
	MiddleSegmentRotated	The text is displayed rotated at the same angle as the segment where the text is placed. A long enough segment from the middle link segments is chosen for that purpose.

Remarks

Values from this enumeration can be assigned to the TextStyle property of links or the LinkTextStyle property of the diagram.

See Also

MindFusion.Diagramming Namespace

LinkValidationEventArgs Class

Contains the arguments passed to handlers of link validation events.

Syntax

C# Copy Code

```
public class LinkValidationEventArgs : LinkEventArgs
```

Visual Basic Copy Code

```
Public Class LinkValidationEventArgs
    Inherits LinkEventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.LinkEventArgs
      MindFusion.Diagramming.LinkValidationEventArgs

See Also

LinkValidationEventArgs Members
MindFusion.Diagramming Namespace

LinkValidationEventArgs Members

Contains the arguments passed to handlers of link validation events.

The following tables list the members exposed by the LinkValidationEventArgs type.

Public Constructors

	Name	Description
	LinkValidationEventArgs	Overloaded. Initializes a new instance of the **LinkValidationEventArgs** class.

Public Properties

	Name	Description
	AdjustmentHandle	Gets or sets the adjustment handle being moved by the user. (Inherited from LinkEventArgs.)
	AnchorIndex	Gets the anchor point related to a link validation event.
	Cancel	Gets or sets a value indicating whether to cancel the current operation.
	ChangingDestination	Gets a value indicating whether the user modifies the link's destination end point.
	ChangingOrigin	Gets a value indicating whether the user modifies the link's origin end point.
	Destination	Gets the node that will become the link's destination if the event is validated.
	Label	Gets or sets the LinkLabel related to this event.

	Link	Gets the link for which the event raised. (Inherited from LinkEventArgs.)
	MouseButton	Gets which mouse button has been pressed. (Inherited from LinkEventArgs.)
	MousePosition	Gets or sets the current mouse pointer position. (Inherited from LinkEventArgs.)
	Node	Gets the node to which the link will be connected.
	NodeElement	Gets the node's element to which the link will be connected.
	Origin	Gets the node that would become the link's origin if the event is validated.
	SelectionRectangle	Gets the current selection rectangle.
	TableRow	Gets the table row related to a link validation event.

Public Methods

	Name	Description
	CancelDrag	Cancels the creation or modification of the link.

See Also

LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs Constructor

Initializes a new instance of the LinkValidationEventArgs class.

Overload List

Name	Description
LinkValidationEventArgs (DiagramLink)	Initializes a new instance of the **LinkValidationEventArgs** class.
LinkValidationEventArgs (DiagramLink, Point, Int32, DiagramNode, Int32, Int32)	Initializes a new instance of the **LinkValidationEventArgs** class.
LinkValidationEventArgs (DiagramLink, Rect)	Initializes a new instance of the **LinkValidationEventArgs** class.

See Also

LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs Constructor (DiagramLink)

Initializes a new instance of the LinkValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkValidationEventArgs (
    DiagramLink link
)
```

Visual Basic Copy Code

```
Public New ( _
    link As DiagramLink _
)
```

Parameters

link
The DiagramLink related to the event.

See Also

LinkValidationEventArgs Constructor Overload List
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs Constructor (DiagramLink, Point, Int32, DiagramNode, Int32, Int32)

Initializes a new instance of the LinkValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkValidationEventArgs (
    DiagramLink link,
    Point mousePosition,
    int adjustmentHandle,
    DiagramNode node,
    int anchorIndex,
    int tableRow
)
```

Visual Basic Copy Code

```
Public New ( _
    link As DiagramLink, _
    mousePosition As Point, _
    adjustmentHandle As Integer, _
    node As DiagramNode, _
    anchorIndex As Integer, _
    tableRow As Integer _
)
```

Parameters

link
The DiagramLink related to the event.

mousePosition
The position of the mouse cursor, in document coordinates.

adjustmentHandle
The zero-based index of the control point related to the event.

node
The DiagramNode related to the event.

anchorIndex
The zero-based index of the anchor point related to the event.

tableRow
The zero-based index of the table row related to the event.

See Also

LinkValidationEventArgs Constructor Overload List
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs Constructor (DiagramLink, Rect)

Initializes a new instance of the LinkValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public LinkValidationEventArgs (
    DiagramLink link,
    Rect selectionRectangle
)
```

Visual Basic Copy Code

```
Public New ( _
    link As DiagramLink, _
    selectionRectangle As Rect _
)
```

Parameters

link
The DiagramLink related to the event.

selectionRectangle
A **Microsoft.Maui.Graphics.Rect** that represents the current selection rectangle.

See Also

LinkValidationEventArgs Constructor Overload List
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs Methods

LinkValidationEventArgs.CancelDrag Method

Cancels the creation or modification of the link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void CancelDrag ()
```

Visual Basic Copy Code

```
Public Sub CancelDrag ()
```

Remarks

Calling this method immediately stops the operation. On the other hand, setting Cancel to **true** only changes the mouse cursor to a Stop icon and the operation is cancelled when the user releases the mouse button.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs Properties

LinkValidationEventArgs.AnchorIndex Property

Gets the anchor point related to a link validation event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int AnchorIndex { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property AnchorIndex As Integer
```

Property Value

An integer value specifying the index of an anchor point within an AnchorPattern.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs.Cancel Property

Gets or sets a value indicating whether to cancel the current operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Cancel { get; set; }
```

Visual Basic Copy Code

```
Public Property Cancel As Boolean
```

Property Value

true to cancel the operation; **false** to approve it.

Remarks

Set this property to **true** to prevent the user from completing the operation.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs.ChangingDestination Property

Gets a value indicating whether the user modifies the link's destination end point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ChangingDestination { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ChangingDestination As Boolean
```

Property Value

true if the user is changing the destination point; otherwise, **false**.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs.ChangingOrigin Property

Gets a value indicating whether the user modifies the link's origin end point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ChangingOrigin { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ChangingOrigin As Boolean
```

Property Value

true if the user is changing the origin point; otherwise, **false**.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs.Destination Property

Gets the node that will become the link's destination if the event is validated.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Destination { get; }
```

Visual Basic

```
Public ReadOnly Property Destination As DiagramNode
```

Property Value

An instance of the DiagramNode class.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs.Label Property

Gets or sets the LinkLabel related to this event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public LinkLabel Label { get; set; }
```

Visual Basic

```
Public Property Label As LinkLabel
```

Property Value

A LinkLabel reference.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs.Node Property

Gets the node to which the link will be connected.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public DiagramNode Node { get; }
```

Visual Basic

```
Public ReadOnly Property Node As DiagramNode
```

Property Value

An instance of the DiagramNode class.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace
NodeElement Property

LinkValidationEventArgs.NodeElement Property

Gets the node's element to which the link will be connected.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object NodeElement { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NodeElement As Object
```

Property Value

The element of a compound node to which the link will connect.

Remarks

This property could refer to a TreeViewItem if the link connects to a TreeViewNode, or a Row if the link connects to a TableNode.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace
Node Property

LinkValidationEventArgs.Origin Property

Gets the node that would become the link's origin if the event is validated.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Origin { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Origin As DiagramNode
```

Property Value

An instance of the DiagramNode class.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs.SelectionRectangle Property

Gets the current selection rectangle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect SelectionRectangle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property SelectionRectangle As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** representing the selection rectangle.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

LinkValidationEventArgs.TableRow Property

Gets the table row related to a link validation event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int TableRow { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property TableRow As Integer
```

Property Value

An integer value specifying the zero-based index of a TableNode's row.

See Also

LinkValidationEventArgs Members
LinkValidationEventArgs Class
MindFusion.Diagramming Namespace

MarkStyle Enumeration

Specifies the visual style of anchor point marks.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum MarkStyle
```

Visual Basic Copy Code

```
Public Enum MarkStyle
```

Members

	Member name	Description

	None	No visual indication for the anchor point.
	Cross	Anchor point marked with a cross.
	X	Anchor point marked with an 'X'.
	Circle	Anchor point marked with a circle.
	Rectangle	Anchor point marked with a rectangle.
	Custom	A DrawAnchorPoint event is fired, allowing custom painting of anchor points.

Remarks

Members of this enumeration are used as values of the MarkStyle property of the AnchorPoint class.

See Also

MindFusion.Diagramming Namespace
AnchorPoint Class
MarkStyle Property
DrawAnchorPoint Event

MeasureUnit Class

Represents a unit of measure.

Syntax

C# Copy Code

```
public class MeasureUnit
```

Visual Basic Copy Code

```
Public Class MeasureUnit
```

Remarks

Instances of this class can be assigned to the MeasureUnit property of the Diagram class to indicate how size and length property values should be mapped to device pixels.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.MeasureUnit

See Also

MeasureUnit Members
MindFusion.Diagramming Namespace

MeasureUnit Members

Represents a unit of measure.

The following tables list the members exposed by the MeasureUnit type.

Public Constructors

	Name	Description
	MeasureUnit	Overloaded.

Public Fields

	Name	Description
	Centimeter	Specifies centimeters as the unit of measure.
	Document	Specifies document units (1/300 inch) as the unit of measure.
	Inch	Specifies inches as the unit of measure.
	Millimeter	Specifies millimeters as the unit of measure.
	Percent	Specifies percents as the unit of measure.
	Pixel	Specifies device pixels as the unit of measure.
	Point	Specifies printer's points (1/72 inch) as the unit of measure.
	WpfPoint	Specifies device-independent pixels (1/96 inch) as the unit of measure.

Public Properties

	Name	Description
	Amount	Gets the unit's size in inches.
	Id	Gets the unit identifier.
	Name	Gets the name of the unit.
	StandardDivisions	Gets the number of subunits that measures of this unit type are commonly divided to.

Public Methods

	Name	Description
	ApplyTo	Overloaded.
	Convert	Overloaded.
	Equals	Overloaded.
	FromGraphicsUnit	Returns the measure unit corresponding to the specified graphics unit.
	FromId	Returns a **MeasureUnit** from its integer identifier.
	GetHashCode	Object.GetHashCode override.
	ToGraphicsUnit	Returns the GraphicsUnit corresponding to this **MeasureUnit**.
	ToString	Object.ToString override.

See Also

MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit Fields

MeasureUnit.Centimeter Field

Specifies centimeters as the unit of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static readonly MeasureUnit Centimeter
```

Visual Basic Copy Code

```
Public Shared ReadOnly Centimeter As MeasureUnit
```

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Document Field

Specifies document units (1/300 inch) as the unit of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static readonly MeasureUnit Document
```

Visual Basic Copy Code

```
Public Shared ReadOnly Document As MeasureUnit
```

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Inch Field

Specifies inches as the unit of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static readonly MeasureUnit Inch
```

Visual Basic Copy Code

```
Public Shared ReadOnly Inch As MeasureUnit
```

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Millimeter Field

Specifies millimeters as the unit of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static readonly MeasureUnit Millimeter
```

Visual Basic Copy Code

```
Public Shared ReadOnly Millimeter As MeasureUnit
```

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Percent Field

Specifies percents as the unit of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static readonly MeasureUnit Percent
```

Visual Basic Copy Code

```
Public Shared ReadOnly Percent As MeasureUnit
```

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Pixel Field

Specifies device pixels as the unit of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static readonly MeasureUnit Pixel
```

Visual Basic Copy Code

```
Public Shared ReadOnly Pixel As MeasureUnit
```

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Point Field

Specifies printer's points (1/72 inch) as the unit of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static readonly MeasureUnit Point
```

Visual Basic Copy Code

```
Public Shared ReadOnly Point As MeasureUnit
```

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.WpfPoint Field

Specifies device-independent pixels (1/96 inch) as the unit of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static readonly MeasureUnit WpfPoint
```

Visual Basic Copy Code

```
Public Shared ReadOnly WpfPoint As MeasureUnit
```

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit Constructor

Overload List

Name	Description
MeasureUnit (Int32, String, MeasureUnitType, Double, Int32)	Initializes one of the built-in units of measure.

MeasureUnit (Int32, String, MeasureUnitType, Double, Int32, IMeasureUnitConverter)	Initializes one of the built-in units of measure.
MeasureUnit (String, Double)	Initializes a new instance of the MeasureUnit class with the specified resolution.
MeasureUnit (String, Double, Int32)	Initializes a new instance of the **MeasureUnit** class with the specified amount.
MeasureUnit (String, MeasureUnit, Double)	Initializes a new instance of the **MeasureUnit** class with the specified unit of measure as a base.
MeasureUnit (String, MeasureUnit, Double, Int32)	Initializes a new instance of the **MeasureUnit** class with the specified unit of measure as a base.

See Also

MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit Constructor (Int32, String, MeasureUnitType, Double, Int32)

Initializes one of the built-in units of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MeasureUnit (
    int id,
    string name,
    MeasureUnitType type,
    double unitsPerInch,
    int standardDivisions
)
```

Visual Basic Copy Code

```
Public New ( _
    id As Integer, _
    name As String, _
    type As MeasureUnitType, _
    unitsPerInch As Double, _
    standardDivisions As Integer _
)
```

Parameters

id
An integer identifier used to represent this unit in saved diagram files.

name
The display name of this unit.

type
A MeasureUnitType enumeration member specifying how unit size should be calculated.

unitsPerInch
A **double** value specifying how many units of this type fit within an inch.

standardDivisions
An integer value specifying how many standard divisions should be displayed for this unit in the scales of the **Ruler** control.

See Also

MeasureUnit Constructor Overload List
MeasureUnit Class

MeasureUnit Constructor (Int32, String, MeasureUnitType, Double, Int32, IMeasureUnitConverter)

Initializes one of the built-in units of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MeasureUnit (
    int id,
    string name,
    MeasureUnitType type,
    double unitsPerInch,
    int standardDivisions,
    IMeasureUnitConverter converter
)
```

Visual Basic Copy Code

```
Public New ( _
    id As Integer, _
    name As String, _
    type As MeasureUnitType, _
    unitsPerInch As Double, _
    standardDivisions As Integer, _
    converter As IMeasureUnitConverter _
)
```

Parameters

id
An integer identifier used to represent this unit in saved diagram files.

name
The display name of this unit.

type
A MeasureUnitType enumeration member specifying how unit size should be calculated.

unitsPerInch
A **double** value specifying how many units of this type fit within an inch.

standardDivisions
An integer value specifying how many standard divisions should be displayed for this unit in the scales of the **Ruler** control.

converter
An IMeasureUnitConverter object used to convert values from this unit to device pixels or other logical units.

See Also

MeasureUnit Constructor Overload List
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit Constructor (String, Double)

Initializes a new instance of the MeasureUnit class with the specified resolution.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MeasureUnit (
    string name,
    double unitsPerInch
)
```

Visual Basic Copy Code

```
Public New ( _
    name As String, _
    unitsPerInch As Double _
)
```

Parameters

name
The name of the new unit.

unitsPerInch
The amount of the new unit that is equal to one inch.

See Also

MeasureUnit Constructor Overload List
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit Constructor (String, Double, Int32)

Initializes a new instance of the MeasureUnit class with the specified amount.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MeasureUnit (
    string name,
    double unitsPerInch,
    int standardDivisions
)
```

Visual Basic Copy Code

```
Public New ( _
    name As String, _
    unitsPerInch As Double, _
    standardDivisions As Integer _
)
```

Parameters

name
The name of the new unit.

unitsPerInch
The amount of the new unit that is equal to one inch.

standardDivisions
The number of divisions of the new unit.

See Also

MeasureUnit Constructor Overload List
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit Constructor (String, MeasureUnit, Double)

Initializes a new instance of the MeasureUnit class with the specified unit of measure as a base.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MeasureUnit (
    string name,
    MeasureUnit baseUnit,
    double scale
)
```

Visual Basic Copy Code

```
Public New ( _
    name As String, _
    baseUnit As MeasureUnit, _
    scale As Double _
)
```

Parameters

name
The name of the new unit.

baseUnit
The base of the new unit.

scale
The scale of the new unit, relatively to the base.

See Also

MeasureUnit Constructor Overload List
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit Constructor (String, MeasureUnit, Double, Int32)

Initializes a new instance of the MeasureUnit class with the specified unit of measure as a base.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MeasureUnit (
    string name,
    MeasureUnit baseUnit,
    double scale,
    int standardDivisions
)
```

Visual Basic Copy Code

```
Public New ( _
    name As String, _
    baseUnit As MeasureUnit, _
    scale As Double, _
    standardDivisions As Integer _
)
```

Parameters

name
The name of the new unit.

baseUnit
The base of the new unit.

scale
The scale of the new unit, relatively to the base.

standardDivisions
The number of divisions of the new unit.

See Also

MeasureUnit Constructor Overload List
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit Methods

MeasureUnit.ApplyTo Method

Overload List

Name	Description
MeasureUnit.ApplyTo (Graphics)	Applies this unit to the specified **Graphics** drawing surface.
MeasureUnit.ApplyTo (IGraphics)	Applies this unit to the specified IGraphics drawing surface.
MeasureUnit.ApplyTo (IMatrix, IGraphics)	Applies this unit to the specified IGraphics drawing surface.

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.ApplyTo Method (Graphics)

Applies this unit to the specified **Graphics** drawing surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ApplyTo (
    Graphics graphics
)
```

Visual Basic Copy Code

```
Public Sub ApplyTo( _
    graphics As Graphics _
)
```

Parameters

graphics
A **Graphics** instance whose **PageUnit** and **PageScale** are set to match the resolution of this unit.

See Also

ApplyTo Method Overload List
MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.ApplyTo Method (IGraphics)

Applies this unit to the specified IGraphics drawing surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ApplyTo (
    IGraphics graphics
)
```

Visual Basic Copy Code

```
Public Sub ApplyTo( _
    graphics As IGraphics _
)
```

Parameters

graphics
An IGraphics instance whose PageUnit and PageScale are set to match the resolution of this unit.

See Also

ApplyTo Method Overload List
MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.ApplyTo Method (IMatrix, IGraphics)

Applies this unit to the specified IGraphics drawing surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ApplyTo (
    IMatrix matrix,
    IGraphics graphics
)
```

Visual Basic Copy Code

```
Public Sub ApplyTo( _
    matrix As IMatrix, _
    graphics As IGraphics _
)
```

Parameters

matrix
An IMatrix instance that should be set to match the resolution of this unit.

graphics
An IGraphics that should supply the DPI.

See Also

ApplyTo Method Overload List
MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Convert Method

Overload List

Name	Description
MeasureUnit.Convert (Double, MeasureUnit, Object)	Converts the specified value from this unit to the specified unit.
MeasureUnit.Convert (Double, MeasureUnit, Object, Axis)	Converts the specified value from this unit to the specified unit.
MeasureUnit.Convert (Point, MeasureUnit, Object)	Converts the specified point from this unit to the specified unit.
MeasureUnit.Convert (Rect, MeasureUnit, Object)	Converts the specified rectangle from this unit to the specified unit.

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Convert Method (Double, MeasureUnit, Object)

Converts the specified value from this unit to the specified unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Convert (
    double value,
    MeasureUnit targetUnit,
    Object context
)
```

Visual Basic Copy Code

```
Public Function Convert( _
    value As Double, _
    targetUnit As MeasureUnit, _
    context As Object _
) As Double
```

Parameters

value
The value to convert, expressed in the current measure unit.

targetUnit
The target measure unit.

context
A parameter that is used when converting from logical units (such as inches and millimeters) to device units (such as pixels) or special units. This can be either **Graphics**, IGraphics, or IResolution.

Return Value
The converted value.

Remarks
If the target represents logical unit (such as millimeter, inch, point) or a unit derived from a logical unit, and the current unit is not logical (for example pixel), the third parameter is required to perform the conversion.

See Also
Convert Method Overload List
MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Convert Method (Double, MeasureUnit, Object, Axis)

Converts the specified value from this unit to the specified unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Convert (
    double value,
    MeasureUnit targetUnit,
    Object context,
    Axis axis
)
```

Visual Basic Copy Code

```
Public Function Convert( _
    value As Double, _
    targetUnit As MeasureUnit, _
    context As Object, _
    axis As Axis _
) As Double
```

Parameters
value
The value to convert, expressed in the current measure unit.

targetUnit
The target measure unit.

context
A parameter that is used when converting from logical units (such as inches and millimeters) to device units (such as pixels) or special units. This can be either **Graphics**, IGraphics, or IResolution.

axis
One of the coordinate axes.

Return Value
The converted value.

Remarks
If the target represents logical unit (such as millimeter, inch, point) or a unit derived from a logical unit, and the current unit is not logical (for example pixel), the third parameter is required to perform the conversion.

See Also

Convert Method Overload List
MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Convert Method (Point, MeasureUnit, Object)

Converts the specified point from this unit to the specified unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point Convert (
    Point p,
    MeasureUnit targetUnit,
    Object context
)
```

Visual Basic Copy Code

```
Public Function Convert( _
    p As Point, _
    targetUnit As MeasureUnit, _
    context As Object _
) As Point
```

Parameters

p
The point to convert, expressed in the current measure unit.

targetUnit
The target measure unit.

context
A parameter that is used when converting from logical units (such as inches and millimeters) to device units (such as pixels) or special units. This can be either **Graphics**, IGraphics, or IResolution.

Return Value

The converted point.

Remarks

If the target represents logical unit (such as millimeter, inch, point) or a unit derived from a logical unit, and the current unit is not logical (for example pixel), the third parameter is required to perform the conversion. If both units are of the same generic type, graphics is ignored.

See Also

Convert Method Overload List
MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Convert Method (Rect, MeasureUnit, Object)

Converts the specified rectangle from this unit to the specified unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect Convert (
    Rect r,
    MeasureUnit targetUnit,
    Object context
)
```

Visual Basic Copy Code

```
Public Function Convert( _
    r As Rect, _
    targetUnit As MeasureUnit, _
    context As Object _
) As Rect
```

Parameters

r
The rectangle to convert, expressed in the current measure unit.

targetUnit
The target measure unit.

context
A parameter that is used when converting from logical units (such as inches and millimeters) to device units (such as pixels) or special units. This can be either **Graphics**, IGraphics, or IResolution.

Return Value

The converted rectangle.

Remarks

If the target represents logical unit (such as mm, inch, point) or a unit derived from a logical unit, and the current unit is not logical (for example pixel), the third parameter is required to perform the conversion. If both units are of the same generic type, graphics is ignored.

See Also

Convert Method Overload List
MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Equals Method

Overload List

Name	Description
MeasureUnit.Equals (MeasureUnit)	Determines whether the specified measure unit is equal to the current measure unit.
MeasureUnit.Equals (Object)	Object.Equals override.

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Equals Method (MeasureUnit)

Determines whether the specified measure unit is equal to the current measure unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public new bool Equals (
    MeasureUnit unit
)
```

Visual Basic Copy Code

```
Public Shadows Function Equals( _
    unit As MeasureUnit _
) As Boolean
```

Parameters

unit
The object to compare to this measure unit.

Return Value

true if the specified measure unit is equal to this object; otherwise, **false**.

See Also

Equals Method Overload List
MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Equals Method (Object)

Object.Equals override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool Equals (
    Object obj
)
```

Visual Basic Copy Code

```
Public Overrides Function Equals( _
    obj As Object _
) As Boolean
```

Parameters

obj
The object to compare with the current unit.

Return Value

true if the specified object is the same as or is equal to the current unit; otherwise, **false**.

See Also

Equals Method Overload List
MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.FromGraphicsUnit Method

Returns the measure unit corresponding to the specified graphics unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static MeasureUnit FromGraphicsUnit (
    GraphicsUnit unit
)
```

Visual Basic Copy Code

```
Public Shared Function FromGraphicsUnit( _
    unit As GraphicsUnit _
) As MeasureUnit
```

Parameters

unit
One of the **GraphicsUnit** values.

Return Value

The built-in MeasureUnit corresponding to the specified graphics unit. If *unit* is **World**, this method returns **null**. If unit is **Display**, this method returns Pixel.

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.FromId Method

Returns a MeasureUnit from its integer identifier.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static MeasureUnit FromId (
    int id
)
```

Visual Basic Copy Code

```
Public Shared Function FromId( _
    id As Integer _
) As MeasureUnit
```

Parameters

id
The identifier of one of the built-in measure units.

Return Value

The built-in MeasureUnit corresponding to the specified identifier.

See Also

MeasureUnit Members
MeasureUnit Class

MeasureUnit.GetHashCode Method

Object.GetHashCode override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override int GetHashCode ()
```

Visual Basic Copy Code

```
Public Overrides Function GetHashCode () As Integer
```

Return Value

A hash code for the current unit.

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.ToGraphicsUnit Method

Returns the GraphicsUnit corresponding to this MeasureUnit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public GraphicsUnit ToGraphicsUnit ()
```

Visual Basic Copy Code

```
Public Function ToGraphicsUnit () As GraphicsUnit
```

Return Value

A member of the GraphicsUnit enumeration if there is one corresponding to this unit, or otherwise **null**.

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.ToString Method

Object.ToString override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override string ToString ()
```

Visual Basic

```
Public Overrides Function ToString () As String
```

Return Value

The name of the unit.

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit Properties

MeasureUnit.Amount Property

Gets the unit's size in inches.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double Amount { get; }
```

Visual Basic

```
Public ReadOnly Property Amount As Double
```

Property Value

A **double** value specifying the size in inches of a single unit of this type.

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Id Property

Gets the unit identifier.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public int Id { get; }
```

Visual Basic

```
Public ReadOnly Property Id As Integer
```

Property Value

Remarks

This property returns the integer identifier passed to MeasureUnit constructors.

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.Name Property

Gets the name of the unit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Name { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Name As String
```

Property Value

The name passed to MeasureUnit constructors.

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnit.StandardDivisions Property

Gets the number of subunits that measures of this unit type are commonly divided to.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int StandardDivisions { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property StandardDivisions As Integer
```

Property Value

The *standardDivisions* value passed to MeasureUnit constructors.

See Also

MeasureUnit Members
MeasureUnit Class
MindFusion.Diagramming Namespace

MeasureUnitType Enumeration

Specifies the type of a unit of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum MeasureUnitType
```

Visual Basic Copy Code

```
Public Enum MeasureUnitType
```

Members

	Member name	Description
	Logical	Indicates logical unit of measure, expressed as units-per-inch.
	Device	Indicates device-dependent unit of measure, expressed in device pixels.
	Special	Indicates special unit of measure, such as percentage.

See Also

MindFusion.Diagramming Namespace

ModificationEffect Enumeration

Specifies the visual effect to apply during interactive modification of items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum ModificationEffect
```

Visual Basic Copy Code

```
Public Enum ModificationEffect
```

Members

	Member name	Description
	None	Show the actual modified item as it would look if the user completes the modification at the current mouse position.
	MoveShades	Show a semi-transparent shade of the modified item at the current mouse position.

See Also

MindFusion.Diagramming Namespace

ModificationStart Enumeration

Specifies how the user can start modifying an item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum ModificationStart
```

Visual Basic Copy Code

```
Public Enum ModificationStart
```

Members

	Member name	Description
	SelectedOnly	Only selected items can be modified.
	AutoHandles	Adjustment handles appear automatically around the item under the mouse; it can be modified without selecting it first.

Remarks

This enumeration specifies the possible values of the **ModificationStart** property.

See Also

MindFusion.Diagramming Namespace
ModificationStart Property

ModifierKeyAction Enumeration

Defines the actions that can be assigned to a modifier key such as CTRL or ALT.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum ModifierKeyAction
```

Visual Basic Copy Code

```
Public Enum ModifierKeyAction
```

Members

	Member name	Description
	None	Specifies that a modifier key does not have any specific function assigned to it.
	Pan	Pan the view if the mouse is dragged while the modifier key is pressed down.
	Select	Start drawing a selection rectangle if the mouse is dragged while a modifier key is pressed down. Toggle the selection if an item is clicked while the key is down.

	OverrideBehavior	While the modifier key is pressed down, dragging the mouse starts creating a new item or drawing a selection rectangle instead of modifying a selected item. Otherwise, dragging the mouse could start modification of the selected item.
	Magnify	Magnifies the diagram objects under the mouse while the modifier key is pressed down.
	ExtendSelection	Similar to **Select**, but does not deselect currently selected items when they are outside of a newly drawn lasso rectangle.
	Zoom	Start drawing a zoom-lasso if the mouse is dragged while respective modifier key is pressed down.

Remarks

The values from this enumeration can be assigned to the members of the diagram view's **ModifierKeyActions**.

See Also

MindFusion.Diagramming Namespace
ModifierKeyActions Class

MouseButton Enumeration

Specifies constants that define which mouse button was pressed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum MouseButton
```

Visual Basic Copy Code

```
Public Enum MouseButton
```

Members

	Member name	Description
	None	No mouse button was pressed.
	Left	The left mouse button was pressed.
	Right	The right mouse button was pressed.
	Middle	The middle mouse button was pressed.
	XButton1	The first XButton was pressed.
	XButton2	The second XButton was pressed.

See Also

MindFusion.Diagramming Namespace

MouseButtonActions Enumeration

Specifies what actions can be associated with the middle and right mouse buttons.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum MouseButtonActions
```

Visual Basic Copy Code

```
Public Enum MouseButtonActions
```

Members

	Member name	Description
	None	Do not associate any action with the button.
	Pan	Pans the view if the mouse is dragged while the button is pressed down.
	Cancel	Cancels the current drawing operation.
	Select	Starts a multiple selection operation.
	Draw	Draw new items or modify existing items.
	Magnify	Magnifies the items under the mouse while the button is pressed down.
	Zoom	Draw a zoom-lasso if the mouse is dragged with specified button pressed down.

Remarks

The **Cancel** value can be combined with **Pan** or **Select**. Thus if a user presses the mouse button while drawing an item, the operation will be cancelled; otherwise dragging with the specified button will start panning or drawing a selection rectangle.

See Also

MindFusion.Diagramming Namespace
MiddleButtonActions Property
RightButtonActions Property

MouseWheelAction Enumeration

Specifies the type of action performed when the user rotates the mouse wheel.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum MouseWheelAction
```

Visual Basic Copy Code

```
Public Enum MouseWheelAction
```

Members

	Member name	Description
	None	The DiagramView control does not handle wheel rotation internally, but raises MouseWheel event to allow custom handling.
	Scroll	Indicates that the mouse wheel should scroll the view.

	Zoom	Indicates that the mouse wheel should zoom the view.

See Also

MindFusion.Diagramming Namespace

NodeAdjustmentHandle Class

Represents an adjustment handle used to move, resize or rotate a node.

Syntax

C# Copy Code

```
public class NodeAdjustmentHandle : AdjustmentHandle
```

Visual Basic Copy Code

```
Public Class NodeAdjustmentHandle
    Inherits AdjustmentHandle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.AdjustmentHandle
    MindFusion.Diagramming.NodeAdjustmentHandle

See Also

NodeAdjustmentHandle Members
MindFusion.Diagramming Namespace

NodeAdjustmentHandle Members

Represents an adjustment handle used to move, resize or rotate a node.

The following tables list the members exposed by the NodeAdjustmentHandle type.

Public Constructors

	Name	Description
	NodeAdjustmentHandle	Initializes a new instance of the **NodeAdjustmentHandle** class.

Public Fields

	Name	Description
	Move	A **NodeAdjustmentHandle** used to move nodes.
	ResizeBottomRight	A **NodeAdjustmentHandle** used to resize nodes from bottom-right corner.

Public Properties

	Name	Description
	HandleType	Gets or sets the type of this adjustment handle.

Public Methods

	Name	Description

	IsMoveHandle	Returns a value indicating whether this handle is used to move nodes.
	IsResizeHandle	Gets whether this adjustment handle resizes the item.
	IsRotationHandle	Gets whether this adjustment handle changes item's rotation angle.

See Also

NodeAdjustmentHandle Class
MindFusion.Diagramming Namespace

NodeAdjustmentHandle Fields

NodeAdjustmentHandle.Move Field

A NodeAdjustmentHandle used to move nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static NodeAdjustmentHandle Move
```

Visual Basic Copy Code

```
Public Shared Move As NodeAdjustmentHandle
```

See Also

NodeAdjustmentHandle Members
NodeAdjustmentHandle Class
MindFusion.Diagramming Namespace

NodeAdjustmentHandle.ResizeBottomRight Field

A NodeAdjustmentHandle used to resize nodes from bottom-right corner.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static NodeAdjustmentHandle ResizeBottomRight
```

Visual Basic Copy Code

```
Public Shared ResizeBottomRight As NodeAdjustmentHandle
```

See Also

NodeAdjustmentHandle Members
NodeAdjustmentHandle Class
MindFusion.Diagramming Namespace

NodeAdjustmentHandle Constructor

Initializes a new instance of the NodeAdjustmentHandle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeAdjustmentHandle (
    NodeHandleType handleType
)
```

Visual Basic Copy Code

```
Public New ( _
    handleType As NodeHandleType _
)
```

Parameters

handleType
A member of the NodeHandleType enumeration identifying the handle type.

See Also

NodeAdjustmentHandle Class
MindFusion.Diagramming Namespace

NodeAdjustmentHandle Methods

NodeAdjustmentHandle.IsMoveHandle Method

Returns a value indicating whether this handle is used to move nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool IsMoveHandle ()
```

Visual Basic Copy Code

```
Public Overrides Function IsMoveHandle () As Boolean
```

Return Value

true if this is a move handle, or **false** otherwise.

See Also

NodeAdjustmentHandle Members
NodeAdjustmentHandle Class
MindFusion.Diagramming Namespace

NodeAdjustmentHandle.IsResizeHandle Method

Gets whether this adjustment handle resizes the item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool IsResizeHandle ()
```

Visual Basic Copy Code

```
Public Overrides Function IsResizeHandle () As Boolean
```

Return Value

true if this handle resizes the item, or false otherwise.

See Also

NodeAdjustmentHandle Members
NodeAdjustmentHandle Class
MindFusion.Diagramming Namespace

NodeAdjustmentHandle.IsRotationHandle Method

Gets whether this adjustment handle changes item's rotation angle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool IsRotationHandle ()
```

Visual Basic Copy Code

```
Public Overrides Function IsRotationHandle () As Boolean
```

Return Value

true if this handle rotates the item, or false otherwise.

See Also

NodeAdjustmentHandle Members
NodeAdjustmentHandle Class
MindFusion.Diagramming Namespace

NodeAdjustmentHandle Properties

NodeAdjustmentHandle.HandleType Property

Gets or sets the type of this adjustment handle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeHandleType HandleType { get; set; }
```

Visual Basic Copy Code

```
Public Property HandleType As NodeHandleType
```

Property Value

A member of the NodeHandleType enumeration specifying the type of this handle.

See Also

NodeAdjustmentHandle Members
NodeAdjustmentHandle Class
MindFusion.Diagramming Namespace

NodeConstraints Class

Defines specific constraints that can be applied to the size and location of nodes.

Syntax

C# Copy Code

```
public class NodeConstraints : ICloneable
```

Visual Basic Copy Code

```
Public Class NodeConstraints
    Implements ICloneable
```

Remarks

Instances of this class can be assigned to the **Constraints** property of nodes.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.NodeConstraints

See Also

NodeConstraints Members
MindFusion.Diagramming Namespace

NodeConstraints Members

Defines specific constraints that can be applied to the size and location of nodes.

The following tables list the members exposed by the NodeConstraints type.

Public Constructors

	Name	Description
	NodeConstraints	Initializes a new instance of the **NodeConstraints** class.

Public Properties

	Name	Description
	DisableMirroring	Gets or sets a value indicating whether the user can drag a side of the node past its opposite side, so that they switch positions.
	KeepInsideDiagram	Gets or sets a value indicating whether a node should always stay inside the boundaries of the diagram that contains it.

	KeepInsideParent	Gets or sets a value indicating whether a node must always stay in the boundaries of its group master node.
	KeepRatio	Gets or sets a value indicating whether the initial width/height ratio of a node must be preserved.
	MaxHeight	Gets or sets the maximal node height allowed.
	MaxWidth	Gets or sets the maximal node width allowed.
	MinHeight	Gets or sets the minimal node height allowed.
	MinWidth	Gets or sets the minimal node width allowed.
	MoveDirection	Gets or sets constraints on movement direction.

Public Methods

	Name	Description
	Clone	Creates a copy of this object.
	LoadFromXml	Loads the label from the specified XML element.
	SaveToXml	Saves the label into the specified XML element.

See Also

NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints Constructor

Initializes a new instance of the NodeConstraints class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeConstraints ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints Methods

NodeConstraints.Clone Method

Creates a copy of this object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As Object
```

Return Value

An **Object** instance that is a copy of the current NodeConstraints instance.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.LoadFromXml Method

Loads the label from the specified XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
A XmlElement instance.

context
A XmlPersistContext instance.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.SaveToXml Method

Saves the label into the specified XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToXml (
    XmlElement xmlElement,
```

```
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

A XmlElement instance.

context

A XmlPersistContext instance.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints Properties

NodeConstraints.DisableMirroring Property

Gets or sets a value indicating whether the user can drag a side of the node past its opposite side, so that they switch positions.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool DisableMirroring { get; set; }
```

Visual Basic Copy Code

```
Public Property DisableMirroring As Boolean
```

Property Value

true to disable mirroring; otherwise **false**. The default value is **false**.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.KeepInsideDiagram Property

Gets or sets a value indicating whether a node should always stay inside the boundaries of the diagram that contains it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool KeepInsideDiagram { get; set; }
```

Visual Basic

```
Public Property KeepInsideDiagram As Boolean
```

Property Value

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.KeepInsideParent Property

Gets or sets a value indicating whether a node must always stay in the boundaries of its group master node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool KeepInsideParent { get; set; }
```

Visual Basic

```
Public Property KeepInsideParent As Boolean
```

Property Value

true if users should be prevented from moving the node out of its parent; otherwise, **false**. The default is **false**.

Remarks

This property lets you restrict nodes attached to another node by calling the AttachTo method so that users cannot move the attached nodes out of the bounds of their parent.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.KeepRatio Property

Gets or sets a value indicating whether the initial width/height ratio of a node must be preserved.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool KeepRatio { get; set; }
```

Visual Basic

```
Public Property KeepRatio As Boolean
```

Property Value

true if the node width to height ratio should be preserved; otherwise, **false**. The default is **false**.

Remarks

If you enable this property in an initialization-event handler, the node will have an aspect ratio of 1:1. In order to let users draw perfect square or circle shapes, set **KeepRatio** to **true** in the InitializeNode handler.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.MaxHeight Property

Gets or sets the maximal node height allowed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double MaxHeight { get; set; }
```

Visual Basic Copy Code

```
Public Property MaxHeight As Double
```

Property Value

0 = no height constraints; any other value specifies the maximal node height allowed.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.MaxWidth Property

Gets or sets the maximal node width allowed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double MaxWidth { get; set; }
```

Visual Basic Copy Code

```
Public Property MaxWidth As Double
```

Property Value

0 = no width constraints; any other value specifies the maximal node width allowed.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.MinHeight Property

Gets or sets the minimal node height allowed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double MinHeight { get; set; }
```

Visual Basic Copy Code

```
Public Property MinHeight As Double
```

Property Value

0 = no height constraints; any other value specifies the minimal node height allowed.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.MinWidth Property

Gets or sets the minimal node width allowed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double MinWidth { get; set; }
```

Visual Basic Copy Code

```
Public Property MinWidth As Double
```

Property Value

0 = no width constraints; any other value specifies the minimal node width allowed.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeConstraints.MoveDirection Property

Gets or sets constraints on movement direction.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DirectionConstraint MoveDirection { get; set; }
```

Visual Basic Copy Code

```
Public Property MoveDirection As DirectionConstraint
```

Property Value

One of the DirectionConstraint enumeration values; the default is **None**.

See Also

NodeConstraints Members
NodeConstraints Class
MindFusion.Diagramming Namespace

NodeEffect Class

Represents the base class of all visual effects that can be applied to the nodes in a Diagram.

Syntax

C# Copy Code

```
public abstract class NodeEffect : INotifyPropertyChanged, ICloneable
```

Visual Basic Copy Code

```
Public MustInherit Class NodeEffect
    Implements INotifyPropertyChanged, ICloneable
```

Remarks

Effects can be applied by adding instances of the concrete effect classes to the NodeEffects collection of the Diagram.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.NodeEffect
MindFusion.Diagramming.AeroEffect
MindFusion.Diagramming.EmbossEffect
MindFusion.Diagramming.GlassEffect
MindFusion.Diagramming.SmoothShadowEffect
MindFusion.Diagramming.AeroEffect
MindFusion.Diagramming.GlassEffect

See Also

NodeEffect Members
MindFusion.Diagramming Namespace

NodeEffect Members

Represents the base class of all visual effects that can be applied to the nodes in a Diagram.

The following tables list the members exposed by the NodeEffect type.

Protected Constructors

	Name	Description
	NodeEffect	Initializes a new instance of the NodeEffect class.

Public Methods

	Name	Description
	Clone	**ICloneable.Clone** implementation.

Public Events

	Name	Description
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation.

See Also

NodeEffect Class
MindFusion.Diagramming Namespace

NodeEffect Constructor

Initializes a new instance of the NodeEffect class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected NodeEffect ()
```

Visual Basic Copy Code

```
Protected Sub New ()
```

See Also

NodeEffect Class
MindFusion.Diagramming Namespace

NodeEffect Methods

NodeEffect.Clone Method

ICloneable.Clone implementation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public abstract Object Clone ()
```

Visual Basic Copy Code

```
Public MustOverride Function Clone () As Object
```

Return Value

A new effect that is a copy of this effect.

See Also

NodeEffect Members
NodeEffect Class
MindFusion.Diagramming Namespace

NodeEffect Events

NodeEventArgs Class

Contains the arguments passed to handlers of node-related events, such as TreeCollapsed and TreeExpanded.

Syntax

C# Copy Code

```
public class NodeEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class NodeEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.NodeEventArgs
      MindFusion.Diagramming.ContainerChildEventArgs
      MindFusion.Diagramming.NodeValidationEventArgs
      MindFusion.Diagramming.ComponentEventArgs
      MindFusion.Diagramming.DrawAnchorPointEventArgs

See Also

NodeEventArgs Members
MindFusion.Diagramming Namespace

NodeEventArgs Members

Contains the arguments passed to handlers of node-related events, such as TreeCollapsed and TreeExpanded.

The following tables list the members exposed by the NodeEventArgs type.

Public Constructors

	Name	Description
	NodeEventArgs	Overloaded. Initializes a new instance of the **NodeEventArgs** class.

Public Properties

	Name	Description
	AdjustmentHandle	Gets or sets the adjustment handle being moved by the user.
	Label	Gets or sets a NodeLabel related to this event.
	MouseButton	Gets which mouse button has been pressed.
	MousePosition	Gets or sets the current mouse pointer position.
	Node	Gets the node for which the event was raised.

See Also

NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs Constructor

Initializes a new instance of the NodeEventArgs class.

Overload List

Name	Description
NodeEventArgs (DiagramNode)	Initializes a new instance of the **NodeEventArgs** class.
NodeEventArgs (DiagramNode, MouseButton, Point)	Initializes a new instance of the **NodeEventArgs** class.
NodeEventArgs (DiagramNode, Point, AdjustmentHandles)	Initializes a new instance of the **NodeEventArgs** class.

See Also

NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs Constructor (DiagramNode)

Initializes a new instance of the NodeEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeEventArgs (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode _
)
```

Parameters

node
The DiagramNode related to the event.

See Also

NodeEventArgs Constructor Overload List
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs Constructor (DiagramNode, MouseButton, Point)

Initializes a new instance of the NodeEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeEventArgs (
    DiagramNode node,
```

```
    MouseButton mouseButton,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode, _
    mouseButton As MouseButton, _
    mousePosition As Point _
)
```

Parameters

node
The DiagramNode related to the event.

mouseButton
The mouse button related to the event.

mousePosition
The position of the mouse cursor, in document coordinates.

See Also

NodeEventArgs Constructor Overload List
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs Constructor (DiagramNode, Point, AdjustmentHandles)

Initializes a new instance of the NodeEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeEventArgs (
    DiagramNode node,
    Point mousePosition,
    AdjustmentHandles adjustmentHandle
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode, _
    mousePosition As Point, _
    adjustmentHandle As AdjustmentHandles _
)
```

Parameters

node
The DiagramNode related to the event.

mousePosition
The position of the mouse cursor, in document coordinates.

adjustmentHandle
The zero-based index of the adjustment handle related to the event.

See Also

NodeEventArgs Constructor Overload List
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs Properties

NodeEventArgs.AdjustmentHandle Property

Gets or sets the adjustment handle being moved by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AdjustmentHandles AdjustmentHandle { get; set; }
```

Visual Basic Copy Code

```
Public Property AdjustmentHandle As AdjustmentHandles
```

Property Value

A member of the AdjustmentHandles enumeration specifying what handle has been used.

See Also

NodeEventArgs Members
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs.Label Property

Gets or sets a NodeLabel related to this event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeLabel Label { get; set; }
```

Visual Basic Copy Code

```
Public Property Label As NodeLabel
```

Property Value

See Also

NodeEventArgs Members
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs.MouseButton Property

Gets which mouse button has been pressed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MouseButton MouseButton { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MouseButton As MouseButton
```

Property Value

A member of the MouseButton enumeration specifying the currently pressed mouse button.

See Also

NodeEventArgs Members
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs.MousePosition Property

Gets or sets the current mouse pointer position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point MousePosition { get; set; }
```

Visual Basic Copy Code

```
Public Property MousePosition As Point
```

Property Value

A **Microsoft.Maui.Graphics.Point** specifying the current mouse position.

See Also

NodeEventArgs Members
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeEventArgs.Node Property

Gets the node for which the event was raised.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Node { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Node As DiagramNode
```

Property Value

An instance of the DiagramNode class.

See Also

NodeEventArgs Members
NodeEventArgs Class
MindFusion.Diagramming Namespace

NodeHandleType Enumeration

Identifies the types of selection handles of a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum NodeHandleType
```

Visual Basic Copy Code

```
Public Enum NodeHandleType
```

Members

	Member name	Description
	ResizeTopLeft	A top-left handle that can be used to resize the node.
	ResizeTopRight	A top-right handle that can be used to resize the node.
	ResizeBottomRight	A bottom-right handle that can be used to resize the node.
	ResizeBottomLeft	A bottom-left handle that can be used to resize the node.
	ResizeTopCenter	A top-center handle that can be used to resize the node vertically.
	ResizeMiddleRight	A middle-right handle that can be used to resize the node horizontally.
	ResizeBottomCenter	A bottom-center handle that can be used to resize the node vertically.
	ResizeMiddleLeft	A middle-left handle that can be used to resize the node horizontally.
	Move	A central selection handle that can be used to move the node.
	Rotate	A rotation handle that can be used to rotate the node.

See Also

MindFusion.Diagramming Namespace

NodeLabel Class

Represents a supplementary label displayed by a DiagramNode. New labels can be added to a node by calling its AddLabel method.

Syntax

C# Copy Code

```
public class NodeLabel : ItemLabel
```

Visual Basic

```
Public Class NodeLabel
    Inherits ItemLabel
```

Remarks

The position of a label is determined by where it is pinned relative to node's bounding box, and offset from the pin point, as set using SetCornerPosition, SetEdgePosition and SetCenterPosition methods. In addition, the HorizontalAlign and VerticalAlign properties specify whether the label is centered on its pin point or aligned to it.

You can let users arrange labels manually by setting the **Behavior** property of **DiagramView** to MoveLabels.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ItemLabel
    MindFusion.Diagramming.NodeLabel

See Also

NodeLabel Members
MindFusion.Diagramming Namespace

NodeLabel Members

Represents a supplementary label displayed by a DiagramNode. New labels can be added to a node by calling its AddLabel method.

The following tables list the members exposed by the NodeLabel type.

Public Constructors

	Name	Description
	NodeLabel	Overloaded.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to draw the label's background. (Inherited from ItemLabel.)
	ContentPadding	Gets or sets the distance between image and text when both are specified. (Inherited from ItemLabel.)
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled. (Inherited from ItemLabel.)
	EnableWrap	Gets or sets a value indicating whether the label's text can be wrapped. (Inherited from ItemLabel.)
	Font	Gets or sets the label's font. (Inherited from ItemLabel.)
	HorizontalAlign	Gets or sets the label's horizontal alignment. (Inherited from ItemLabel.)
	HorizontalOffset	Gets or sets the label's horizontal offset from its pin point. (Inherited from ItemLabel.)
	Image	Gets or sets the image displayed by this label. (Inherited from ItemLabel.)
	ImageAlign	Gets or sets the alignment of label's image relative to text. (Inherited from ItemLabel.)
	ImageSize	Gets or sets a size that overrides the native image size. (Inherited from ItemLabel.)
	ImageUrl	Gets or sets URL of the image to display inside this label. (Inherited from ItemLabel.)

	Index	Gets or sets the index of an edge or point this label is relative to. (Inherited from ItemLabel.)
	Item	Gets or sets the DiagramItem that displays this label. (Inherited from ItemLabel.)
	Locked	Gets or sets a value indicating whether users are allowed to edit this label. (Inherited from ItemLabel.)
	Margin	Gets or sets the label's margin. (Inherited from ItemLabel.)
	MaxWidth	Gets or sets the maximum width of the label. (Inherited from ItemLabel.)
	Node	Gets or sets the **DiagramNode** that displays this label.
	RelativeTo	Gets or sets the label's relative positioning.
	Text	Gets or sets the label's text. (Inherited from ItemLabel.)
	TextBrush	Gets or sets the brush used to draw the label's text. (Inherited from ItemLabel.)
	VerticalAlign	Gets or sets the label's vertical alignment. (Inherited from ItemLabel.)
	VerticalOffset	Gets or sets the label's vertical offset from its pin point. (Inherited from ItemLabel.)

Public Methods

	Name	Description
	Cancel	Called when drag operation is canceled. (Inherited from ItemLabel.)
	Clone	Creates a clone of this label.
	Draw	Draws the label on the specified IGraphics surface. (Inherited from ItemLabel.)
	GetEditRect	InplaceEditable.GetEditRect implementation. (Inherited from ItemLabel.)
	GetTextToEdit	InplaceEditable.GetTextToEdit implementation. (Inherited from ItemLabel.)
	LoadFromXml	Loads the label from the specified XML element.
	OnMouseDown	Called when the left mouse button is pressed while the pointer is over this label. (Inherited from ItemLabel.)
	OnMouseMove	Called when the mouse is moved while this label tracks the mouse events. (Inherited from ItemLabel.)
	OnMouseUp	Called when the left mouse button is released while this label tracks the mouse events. (Inherited from ItemLabel.)
	SaveToXml	Saves the label into the specified XML element.
	SetCenterPosition	Positions this label relatively to the node center.
	SetCornerPosition	Positions this label relatively to the specified corner point.
	SetEdgePosition	Positions this label relatively to the specified node edge.
	SetEditedText	InplaceEditable.SetEditedText implementation. (Inherited from ItemLabel.)
	SetMouseCursor	Indicates what cursor to display for interactive labels. (Inherited from ItemLabel.)
	UpdateValue	Called while drag operation is in progress. (Inherited from ItemLabel.)

Protected Methods

	Name	Description

	LoadFromJson	Loads the label from a JsonObject.
	SaveToJson	Saves the label into a **JsonObject**.

See Also

NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel Constructor

Overload List

Name	Description
NodeLabel ()	Initializes a new instance of the NodeLabel class.
NodeLabel (DiagramNode)	Initializes a new instance of the **NodeLabel** class.
NodeLabel (DiagramNode, String)	Initializes a new instance of the **NodeLabel** class.

See Also

NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel Constructor ()

Initializes a new instance of the NodeLabel class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeLabel ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

NodeLabel Constructor Overload List
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel Constructor (DiagramNode)

Initializes a new instance of the NodeLabel class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeLabel (
    DiagramNode node
```

```
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode _
)
```

Parameters

node
The DiagramNode that displays the label.

See Also

NodeLabel Constructor Overload List
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel Constructor (DiagramNode, String)

Initializes a new instance of the NodeLabel class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeLabel (
    DiagramNode node,
    string text
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode, _
    text As String _
)
```

Parameters

node
The DiagramNode that displays the label.

text
The label's text.

See Also

NodeLabel Constructor Overload List
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel Methods

NodeLabel.Clone Method

Creates a clone of this label.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeLabel Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As NodeLabel
```

Return Value

A new NodeLabel instance that is an exact copy of this label.

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel.LoadFromJson Method

Loads the label from a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the label's data.

context
A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel.LoadFromXml Method

Loads the label from the specified XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** instance.

context
An XmlPersistContext instance.

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel.SaveToJson Method

Saves the label into a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject the label's data should be stored into.

context
A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel.SaveToXml Method

Saves the label into the specified XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** instance.

context
An XmlPersistContext instance.

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel.SetCenterPosition Method

Positions this label relatively to the node center.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetCenterPosition (
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Public Sub SetCenterPosition( _
    dx As Double, _
    dy As Double _
)
```

Parameters

dx
Horizontal offset from the center point.

dy
Vertical offset from the center point.

Remarks

This method sets the node's center as label's pin point and assigns specified offsets to HorizontalOffset and VerticalOffset properties.

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel.SetCornerPosition Method

Positions this label relatively to the specified corner point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetCornerPosition (
    int cornerPoint,
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Public Sub SetCornerPosition( _
    cornerPoint As Integer, _
    dx As Double, _
    dy As Double _
)
```

Parameters

cornerPoint
The corner point index.

dx
Horizontal offset from the point.

dy
Vertical offset from the point.

Remarks

This method sets node's corner as label's pin point and assigns specified offsets to HorizontalOffset and VerticalOffset properties. The *cornerPoint* index starts counting from top-left corner in clockwise direction:

- 0 desigates top-left corner;
- 1 desigates top-right corner;
- 2 desigates bottom-right corner;
- 3 desigates bottom-left corner;

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel.SetEdgePosition Method

Positions this label relatively to the specified node edge.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetEdgePosition (
    int edge,
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Public Sub SetEdgePosition( _
    edge As Integer, _
    dx As Double, _
    dy As Double _
)
```

Parameters

edge
The edge index.

dx
Horizontal offset from the edge center.

dy
Vertical offset from the edge center.

Remarks

This method sets the middle point of node's edge as label's pin point and assigns specified offsets to HorizontalOffset and VerticalOffset properties. The edge index starts counting from top side in clockwise direction:

- 0 desigates center of top edge;
- 1 desigates center of right edge;
- 2 desigates center of bottom edge;
- 3 desigates center of left edge;

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeLabel Properties

NodeLabel.Node Property

Gets or sets the DiagramNode that displays this label.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Node { get; set; }
```

Visual Basic Copy Code

```
Public Property Node As DiagramNode
```

Property Value

A DiagramNode reference.

See Also

NodeLabel Members
NodeLabel Class

NodeLabel.RelativeTo Property

Gets or sets the label's relative positioning.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RelativeToNode RelativeTo { get; set; }
```

Visual Basic Copy Code

```
Public Property RelativeTo As RelativeToNode
```

Property Value

A member of the RelativeToNode enumeration. The default is **Center**.

See Also

NodeLabel Members
NodeLabel Class
MindFusion.Diagramming Namespace

NodeValidationEventArgs Class

Contains the arguments passed to handlers of node validation events.

Syntax

C# Copy Code

```
public class NodeValidationEventArgs : NodeEventArgs
```

Visual Basic Copy Code

```
Public Class NodeValidationEventArgs
    Inherits NodeEventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.NodeEventArgs
      MindFusion.Diagramming.NodeValidationEventArgs

See Also

NodeValidationEventArgs Members
MindFusion.Diagramming Namespace

NodeValidationEventArgs Members

Contains the arguments passed to handlers of node validation events.

The following tables list the members exposed by the NodeValidationEventArgs type.

Public Constructors

	Name	Description
	NodeValidationEventArgs	Overloaded. Initializes a new instance of the **NodeValidationEventArgs** class.

Public Properties

	Name	Description
	AdjustmentHandle	Gets or sets the adjustment handle being moved by the user. (Inherited from NodeEventArgs.)
	Cancel	Gets or sets a value indicating whether to cancel the current operation.
	Label	Gets or sets a NodeLabel related to this event.
	MouseButton	Gets which mouse button has been pressed. (Inherited from NodeEventArgs.)
	MousePosition	Gets or sets the current mouse pointer position. (Inherited from NodeEventArgs.)
	Node	Gets the node for which the event was raised. (Inherited from NodeEventArgs.)
	OverrideBounds	Gets or sets a rectangle that should be assigned to the **Bounds** property of nodes after NodeModifying event is raised.
	SelectionRectangle	Gets the current selection rectangle.

Public Methods

	Name	Description
	CancelDrag	Cancels the creation or modification of the node.
	ShouldCancelDrag	Gets whether the interaction should cancel immediately.

See Also

NodeValidationEventArgs Class
MindFusion.Diagramming Namespace

NodeValidationEventArgs Constructor

Initializes a new instance of the NodeValidationEventArgs class.

Overload List

Name	Description
NodeValidationEventArgs (DiagramNode)	Initializes a new instance of the NodeValidationEventArgs class.
NodeValidationEventArgs (DiagramNode, Point, AdjustmentHandles)	Initializes a new instance of the **NodeValidationEventArgs** class.
NodeValidationEventArgs (DiagramNode, Rect)	Initializes a new instance of the **NodeValidationEventArgs** class.

See Also

NodeValidationEventArgs Class
MindFusion.Diagramming Namespace

NodeValidationEventArgs Constructor (DiagramNode)

Initializes a new instance of the NodeValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeValidationEventArgs (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode _
)
```

Parameters

node
The DiagramNode related to the event.

See Also

NodeValidationEventArgs Constructor Overload List
NodeValidationEventArgs Class
MindFusion.Diagramming Namespace

NodeValidationEventArgs Constructor (DiagramNode, Point, AdjustmentHandles)

Initializes a new instance of the NodeValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeValidationEventArgs (
    DiagramNode node,
    Point mousePosition,
    AdjustmentHandles adjustmentHandle
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode, _
    mousePosition As Point, _
    adjustmentHandle As AdjustmentHandles _
)
```

Parameters

node
The DiagramNode related to the event.

mousePosition
The position of the mouse cursor, in document coordinates.

adjustmentHandle
The zero-based index of the adjustment handle related to the event.

See Also

NodeValidationEventArgs Constructor Overload List
NodeValidationEventArgs Class
MindFusion.Diagramming Namespace

NodeValidationEventArgs Constructor (DiagramNode, Rect)

Initializes a new instance of the NodeValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeValidationEventArgs (
    DiagramNode node,
    Rect selectionRectangle
)
```

Visual Basic Copy Code

```
Public New ( _
    node As DiagramNode, _
    selectionRectangle As Rect _
)
```

Parameters

node
The DiagramNode related to the event.

selectionRectangle
A **Microsoft.Maui.Graphics.Rect** representing the selection rectangle.

See Also

NodeValidationEventArgs Constructor Overload List
NodeValidationEventArgs Class
MindFusion.Diagramming Namespace

NodeValidationEventArgs Methods

NodeValidationEventArgs.CancelDrag Method

Cancels the creation or modification of the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void CancelDrag ()
```

Visual Basic Copy Code

```
Public Sub CancelDrag ()
```

Remarks

Calling this method immediately stops the operation. On the other hand, setting Cancel to **true** only changes the mouse cursor to a Stop icon and the operation is cancelled when the user releases the mouse button.

See Also

NodeValidationEventArgs Members
NodeValidationEventArgs Class
MindFusion.Diagramming Namespace

NodeValidationEventArgs.ShouldCancelDrag Method

Gets whether the interaction should cancel immediately.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ShouldCancelDrag ()
```

Visual Basic Copy Code

```
Public Function ShouldCancelDrag () As Boolean
```

Return Value

true to cancel, or false otherwise.

See Also

NodeValidationEventArgs Members
NodeValidationEventArgs Class
MindFusion.Diagramming Namespace

NodeValidationEventArgs Properties

NodeValidationEventArgs.Cancel Property

Gets or sets a value indicating whether to cancel the current operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Cancel { get; set; }
```

Visual Basic Copy Code

```
Public Property Cancel As Boolean
```

Property Value

true to cancel the operation; **false** to approve it.

Remarks

Set this property to **true** to prevent the user from completing the operation.

See Also

NodeValidationEventArgs.Label Property

Gets or sets a NodeLabel related to this event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeLabel Label { get; set; }
```

Visual Basic Copy Code

```
Public Property Label As NodeLabel
```

Property Value

A NodeLabel instance.

Remarks

This property is set when events are raised for supplementary node labels, e.g. NodeTextEditing firing when user tries to edit a label.

See Also

NodeValidationEventArgs Members
NodeValidationEventArgs Class
MindFusion.Diagramming Namespace
DiagramNode.AddLabel Method

NodeValidationEventArgs.OverrideBounds Property

Gets or sets a rectangle that should be assigned to the **Bounds** property of nodes after NodeModifying event is raised.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect OverrideBounds { get; set; }
```

Visual Basic Copy Code

```
Public Property OverrideBounds As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** instance. The default value is **Empty**, signifying that the control should apply standard interaction logic when calculating new coordinates.

See Also

NodeValidationEventArgs.SelectionRectangle Property

Gets the current selection rectangle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect SelectionRectangle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property SelectionRectangle As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** representing the selection rectangle.

See Also

NodeValidationEventArgs Members
NodeValidationEventArgs Class
MindFusion.Diagramming Namespace

NotATreeException Class

An exception raised by the TreeEnumerator class if the specified node does not represent a root of a tree.

Syntax

C# Copy Code

```
public class NotATreeException : Exception
```

Visual Basic Copy Code

```
Public Class NotATreeException
    Inherits Exception
```

Inheritance Hierarchy

System.Object
  System.Exception
    MindFusion.Diagramming.NotATreeException

See Also

NotATreeException Members
MindFusion.Diagramming Namespace

NotATreeException Members

An exception raised by the TreeEnumerator class if the specified node does not represent a root of a tree.

The following tables list the members exposed by the NotATreeException type.

Public Constructors

	Name	Description
	NotATreeException	Initializes a new instance of the **NotATreeException** class.

See Also

NotATreeException Class
MindFusion.Diagramming Namespace

NotATreeException Constructor

Initializes a new instance of the NotATreeException class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NotATreeException ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

NotATreeException Class
MindFusion.Diagramming Namespace

NotifyCollectionChangedAction Enumeration

Describes the action that caused a CollectionChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum NotifyCollectionChangedAction
```

Visual Basic Copy Code

```
Public Enum NotifyCollectionChangedAction
```

Members

	Member name	Description
	Add	One or more items were added to the collection.
	Remove	One or more items were removed from the collection.
	Replace	One or more items were replaced in the collection.
	Reset	The content of the collection changed dramatically.

See Also

MindFusion.Diagramming Namespace

NotifyCollectionChangedEventArgs(T) Class

Provides data for the ObservableCollection<T>.CollectionChanged event.

Syntax

C# Copy Code

```
public class NotifyCollectionChangedEventArgs<T> : EventArgs
```

Visual Basic Copy Code

```
Public Class NotifyCollectionChangedEventArgs(Of T)
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.NotifyCollectionChangedEventArgs(T)

See Also

NotifyCollectionChangedEventArgs(T) Members
MindFusion.Diagramming Namespace

NotifyCollectionChangedEventArgs(T) Members

Provides data for the ObservableCollection<T>.CollectionChanged event.

The following tables list the members exposed by the NotifyCollectionChangedEventArgs(T) type.

Public Constructors

	Name	Description
	NotifyCollectionChangedEventArgs	Initializes a new instance of the **NotifyCollectionChangedEventArgs** class.

Public Properties

	Name	Description
	Action	Gets or sets the action that caused the event.
	NewItems	Gets the list of new items involved in the change.
	OldItems	Gets the list of items affected by a Replace, **Remove**, or **Reset** action.

See Also

NotifyCollectionChangedEventArgs(T) Class
MindFusion.Diagramming Namespace

NotifyCollectionChangedEventArgs(T) Constructor

Initializes a new instance of the NotifyCollectionChangedEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NotifyCollectionChangedEventArgs<T> ()
```

Visual Basic Copy Code

```
Public Sub New(Of T) ()
```

See Also

NotifyCollectionChangedEventArgs(T) Class
MindFusion.Diagramming Namespace

NotifyCollectionChangedEventArgs(T) Properties

NotifyCollectionChangedEventArgs(T).Action Property

Gets or sets the action that caused the event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NotifyCollectionChangedAction Action { get; set; }
```

Visual Basic Copy Code

```
Public Property Action As NotifyCollectionChangedAction
```

Property Value

A member of the NotifyCollectionChangedAction enumeration.

See Also

NotifyCollectionChangedEventArgs(T) Members
NotifyCollectionChangedEventArgs(T) Class
MindFusion.Diagramming Namespace

NotifyCollectionChangedEventArgs(T).NewItems Property

Gets the list of new items involved in the change.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public T[] NewItems { get; set; }
```

Visual Basic Copy Code

```
Public Property NewItems As T()
```

Property Value

A list with the new items added to the collection. This can be **null** (**Nothing** in Visual Basic).

See Also

NotifyCollectionChangedEventArgs(T) Members
NotifyCollectionChangedEventArgs(T) Class
MindFusion.Diagramming Namespace

NotifyCollectionChangedEventArgs(T).OldItems Property

Gets the list of items affected by a Replace, **Remove**, or **Reset** action.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public T[] OldItems { get; set; }
```

Visual Basic Copy Code

```
Public Property OldItems As T()
```

Property Value

A list with the items removed from the collection. This can be **null** (**Nothing** in Visual Basic).

See Also

NotifyCollectionChangedEventArgs(T) Members
NotifyCollectionChangedEventArgs(T) Class
MindFusion.Diagramming Namespace

ObservableCollection(T) Class

Represents a dynamic data collection that provides notifications when items get added, removed, or when the whole list is refreshed.

Syntax

C# Copy Code

```
public class ObservableCollection<T> : Collection<T>, INotifyCollectionChanged
```

Visual Basic Copy Code

```
Public Class ObservableCollection(Of T)
    Inherits Collection(Of T)
    Implements INotifyCollectionChanged
```

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.Collection(T)
    MindFusion.Diagramming.ObservableCollection(T)

See Also

ObservableCollection(T) Members
MindFusion.Diagramming Namespace

ObservableCollection(T) Members

Represents a dynamic data collection that provides notifications when items get added, removed, or when the whole list is refreshed.

The following tables list the members exposed by the ObservableCollection(T) type.

Public Constructors

	Name	Description
	ObservableCollection	Overloaded. Initializes a new instance of the **ObservableCollection<T>** class.

Public Methods

	Name	Description
	AddRange	Adds the specified objects to the end of the collection.

Protected Methods

	Name	Description
	ClearItems	**Collection<T>.ClearItems** override.
	InsertItem	**Collection<T>.InsertItem** override.
	OnCollectionChanged	Raises the CollectionChanged event.
	RemoveItem	**Collection<T>.RemoveItem** override.
	SetItem	**Collection<T>.SetItem** override.

Public Events

	Name	Description
	CollectionChanged	Occurs when an item is added, removed, changed, moved, or the entire list is refreshed.

See Also

ObservableCollection(T) Class
MindFusion.Diagramming Namespace

ObservableCollection(T) Constructor

Initializes a new instance of the ObservableCollection<T> class.

Overload List

Name	Description
ObservableCollection(T) ()	Initializes a new instance of the **ObservableCollection<T>** class.
ObservableCollection(T) (ICollection(T))	Initializes a new instance of the **ObservableCollection<T>** class.

See Also

ObservableCollection(T) Class
MindFusion.Diagramming Namespace

ObservableCollection(T) Constructor ()

Initializes a new instance of the ObservableCollection<T> class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ObservableCollection<T> ()
```

Visual Basic Copy Code

```
Public Sub New(Of T) ()
```

See Also

ObservableCollection(T) Constructor Overload List
ObservableCollection(T) Class
MindFusion.Diagramming Namespace

ObservableCollection(T) Constructor (ICollection(T))

Initializes a new instance of the ObservableCollection<T> class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ObservableCollection<T> (
    ICollection<T> items
)
```

Visual Basic Copy Code

```
Public New (Of T) ( _
    items As ICollection(Of T) _
)
```

Parameters

items
The initial contents of the collection.

See Also

ObservableCollection(T) Constructor Overload List
ObservableCollection(T) Class
MindFusion.Diagramming Namespace

ObservableCollection(T) Methods

ObservableCollection(T).AddRange Method

Adds the specified objects to the end of the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void AddRange (
    ICollection<T> items
)
```

Visual Basic

```
Public Sub AddRange( _
    items As ICollection(Of T) _
)
```

Parameters

items
The objects to add.

See Also

ObservableCollection(T) Members
ObservableCollection(T) Class
MindFusion.Diagramming Namespace

ObservableCollection(T).ClearItems Method

Collection<T>.ClearItems override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void ClearItems ()
```

Visual Basic

```
Protected Overrides Sub ClearItems ()
```

See Also

ObservableCollection(T) Members
ObservableCollection(T) Class
MindFusion.Diagramming Namespace

ObservableCollection(T).InsertItem Method

Collection<T>.InsertItem override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void InsertItem (
    int index,
    T item
)
```

Visual Basic

```
Protected Overrides Sub InsertItem( _
    index As Integer, _
    item As T _
)
```

Parameters

index
The zero-based index at which item should be inserted.

item
The object to insert. The value can be **null** (**Nothing** in Visual Basic) for reference types.

See Also

ObservableCollection(T) Members
ObservableCollection(T) Class
MindFusion.Diagramming Namespace

ObservableCollection(T).OnCollectionChanged Method

Raises the CollectionChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnCollectionChanged (
    NotifyCollectionChangedEventArgs<T> e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnCollectionChanged( _
    e As NotifyCollectionChangedEventArgs(Of T) _
)
```

Parameters

e
An object, containing the event data.

See Also

ObservableCollection(T) Members
ObservableCollection(T) Class
MindFusion.Diagramming Namespace

ObservableCollection(T).RemoveItem Method

Collection<T>.RemoveItem override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RemoveItem (
    int index
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RemoveItem( _
    index As Integer _
)
```

Parameters

index
The zero-based index of the element to remove.

See Also

ObservableCollection(T) Members
ObservableCollection(T) Class
MindFusion.Diagramming Namespace

ObservableCollection(T).SetItem Method

Collection<T>.SetItem override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SetItem (
    int index,
    T item
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SetItem( _
    index As Integer, _
    item As T _
)
```

Parameters

index
The zero-based index of the element to replace.

item
The new value for the element at the specified index. The value can be **null** (**Nothing** in Visual Basic) for reference types.

See Also

ObservableCollection(T) Members
ObservableCollection(T) Class
MindFusion.Diagramming Namespace

ObservableCollection(T) Events

ObservableCollection(T).CollectionChanged Event

Occurs when an item is added, removed, changed, moved, or the entire list is refreshed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event EventHandler<NotifyCollectionChangedEventArgs<T>> CollectionChanged
```

Visual Basic Copy Code

```
Public Event CollectionChanged As EventHandler(Of NotifyCollectionChangedEventArgs(Of T))
```

Event Data

CollectionChanged event handlers receive an argument of type NotifyCollectionChangedEventArgs<T>.

See Also

ObservableCollection(T) Members
ObservableCollection(T) Class
MindFusion.Diagramming Namespace

Orientation Enumeration

Defines values that specify the orientation of the first segment of cascading links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum Orientation
```

Visual Basic Copy Code

```
Public Enum Orientation
```

Members

	Member name	Description
	Auto	MvcDiagram determines automatically the initial direction of link's first segment.
	Horizontal	The first segment of the link is always horizontal.
	Vertical	The first segment of the link is always vertical.

See Also

MindFusion.Diagramming Namespace
CascadeOrientation Property

PagedContainerNode Class

A container that can arrange its children in pages.

Syntax

C# Copy Code

```
public class PagedContainerNode : ContainerNode,
     InplaceEditableList
```

Visual Basic Copy Code

```
Public Class PagedContainerNode
    Inherits ContainerNode
    Implements InplaceEditableList
```

Remarks

PagedContainerNode is a ContainerNode subclass that organizes its child nodes into a collection of pages, represented by ContainerPage objects. This allows for the creation of more orgainzed diagrams where nodes can be grouped into logical views within a single container. Only the nodes belonging to the currently active page, specified by CurrentPage property, are rendered and participate in user interactions.

Navigation between pages is handled by scroll arrows in the caption bar, allowing users to cycle through the pages sequentially. For faster access, users can also double-click the container's caption to open an in-place combo box listing available page titles,

enabling direct navigation to any page. The node's Caption automatically displays the Title of newly activated **ContainerPage**.

The appearance of pager arrows can be fully customized by implementing a PageIconRenderer and assigning it to the static PageIconRenderer property of **Renderers**, allowing for custom drawing logic, images, or SVG icons.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.ContainerNode
        MindFusion.Diagramming.PagedContainerNode

See Also

PagedContainerNode Members
MindFusion.Diagramming Namespace

PagedContainerNode Members

A container that can arrange its children in pages.

The following tables list the members exposed by the PagedContainerNode type.

Public Constructors

	Name	Description
	PagedContainerNode	Overloaded.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowAddChildren	Gets or sets a value indicating whether child nodes can be added interactively to this container. (Inherited from ContainerNode.)
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AllowRemoveChildren	Gets or sets a value indicating whether child nodes can be removed interactively from this container. (Inherited from ContainerNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	AutoDeleteChildren	Gets or sets whether deleting this container should also remove its child items from the diagram. (Inherited from ContainerNode.)
	AutoGrow	Gets or sets a value indicating whether the container should grow in size when nodes are added to it. (Inherited from ContainerNode.)
	AutoShrink	Gets or sets a value indicating whether the container should shrink in size when nodes are removed from it. (Inherited from ContainerNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)

	Caption	Gets or sets the text displayed as caption of the table. (Inherited from ContainerNode.)
	CaptionBackBrush	Gets or sets the brush used to paint the caption bar of this container. (Inherited from ContainerNode.)
	CaptionBrush	Gets or sets the brush used to paint the container caption. (Inherited from ContainerNode.)
	CaptionFormat	Gets or sets the caption alignment and format. (Inherited from ContainerNode.)
	CaptionHeight	Gets or sets the height of the container's caption bar. (Inherited from ContainerNode.)
	Children	Gets the child nodes of this container. (Inherited from ContainerNode.)
	ClipChildren	Gets or sets a value indicating whether child items should be clipped by container boundaries. (Inherited from ContainerNode.)
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent **ContainerNode** containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	CurrentPage	Gets or sets the current page.
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveCaptionBackBrush	Gets the effective caption back brush of the container. (Inherited from ContainerNode.)
	EffectiveCaptionBrush	Gets the effective caption brush of the container. (Inherited from ContainerNode.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled. (Inherited from ContainerNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Foldable	Gets or sets a value indicating whether the user is allowed to fold this container. (Inherited from ContainerNode.)
	Folded	Gets or sets a value indicating whether this container is folded. (Inherited from ContainerNode.)
	FoldedSize	Gets or sets the size of the container when folded. (Inherited from ContainerNode.)
	FoldIconSize	Gets or sets the size of the fold/unfold icon. (Inherited from ContainerNode.)

	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HighlightPen	Gets or sets the pen used to draw the container frame lines when an item is dragged over the container. (Inherited from ContainerNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	Margin	Gets or sets how much space should be left between contained nodes and the container borders. (Inherited from ContainerNode.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	MinimumSize	Gets or sets the minimum size of the container. (Inherited from ContainerNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Pages	Gets the pages of this container.
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Scrollable	Gets or sets a value indicating whether users are allowed to scroll the contents of this container. (Inherited from ContainerNode.)

	ScrollX	Gets or sets the container's horizontal scroll position. (Inherited from ContainerNode.)
	ScrollY	Gets or sets the container's vertical scroll position. (Inherited from ContainerNode.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Shape	Gets or sets the shape of the container's outline. (Inherited from ContainerNode.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	UnfoldedSize	Returns the original size of a folded container. (Inherited from ContainerNode.)
	Visible	Gets or sets a value indicating whether the container is visible. (Inherited from ContainerNode.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description

	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	Add	Adds a node to this container. (Inherited from ContainerNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	AddPage	Adds a page to the container.
	Animate	Animates the item. (Inherited from DiagramItem.)
	Arrange	Arranges the container's children with the given layout. (Inherited from ContainerNode.)
	ArrangeAnimated	Arranges the container's children using the given layout, and animates items moving to their new positions. (Inherited from ContainerNode.)
	ArrangeAsync	Arranges the container's children with the given layout asynchronously. (Inherited from ContainerNode.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	Clone	Creates a copy of this item.
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	ContainsRecursively	Determines whether this container contains the specified node, either directly or recursively through other containers. (Inherited from ContainerNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramNode.Draw override. Draws the container on the specified IGraphics surface. (Inherited from ContainerNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	DiagramNode.DrawLocal override. Draws the container on the specified IGraphics surface. (Inherited from ContainerNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified **IGraphics** surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)

	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
S	GetContainer	Retrieves the container node (if any) the specified node belongs to. (Inherited from ContainerNode.)
	GetCurrentItem	Implements InplaceEditableList.GetCurrentItem.
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectiveCaptionBackBrush	Returns a Brush object representing the currently effective caption background of the container. (Inherited from ContainerNode.)
	GetEffectiveCaptionBrush	Returns a Brush object representing the currently effective caption brush of the container. (Inherited from ContainerNode.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetListItems	Implements InplaceEditableList.GetListItems.
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramNode.GetRepaintRect override. (Inherited from ContainerNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	DiagramItem.HitTest override. (Inherited from ContainerNode.)
	HitTestHandle	DiagramNode.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from ContainerNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	InsertPage	Inserts a page at the specified index.
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	MovePage	Moves a page to a new position in the page list.

	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from ContainerNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from ContainerNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from ContainerNode.)
	OnLoad	DiagramItem.OnLoad override. (Inherited from ContainerNode.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	Remove	Removes a node from this container. (Inherited from ContainerNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	RemovePage	Removes a page from the container.
	Resize	Resizes the node. (Inherited from DiagramNode.)
	ResizeToFitChildren	Resizes the container making it big enough to contain its child nodes. (Inherited from ContainerNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetCurrentItem	Implements InplaceEditableList.SetCurrentItem.
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	UpdateBounds	Overloaded. (Inherited from ContainerNode.)
	VisitNodes	Visits the diagram nodes in reverse Z order. (Inherited from ContainerNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from ContainerNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the item content from a JsonObject.
	LoadFromXml	Loads the item content from an XML element.
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from ContainerNode.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramNode.OnRemove override. (Inherited from ContainerNode.)

	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramNode.RestoreProperties override. (Inherited from ContainerNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramNode.SaveProperties override. (Inherited from ContainerNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the item content into a **JsonObject**.
	SaveToXml	Saves the item content into an XML element.
	ScaleElements	DiagramNode.ScaleElements override. (Inherited from ContainerNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)

	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode Constructor

Overload List

Name	Description
PagedContainerNode ()	Initializes a new instance of the PagedContainerNode class.
PagedContainerNode (Diagram)	Initializes a new instance of the **PagedContainerNode** class.
PagedContainerNode (Diagram, IEnumerable(String))	Initializes a new instance of the **PagedContainerNode** class with the specified page titles.
PagedContainerNode (PagedContainerNode)	Initializes a new instance of the **PagedContainerNode** class using the specified node as a prototype.

See Also

PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode Constructor ()

Initializes a new instance of the PagedContainerNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PagedContainerNode ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

PagedContainerNode Constructor Overload List
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode Constructor (Diagram)

Initializes a new instance of the PagedContainerNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PagedContainerNode (
    Diagram parent
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram _
)
```

Parameters

parent
The parent Diagram instance.

See Also

PagedContainerNode Constructor Overload List
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode Constructor (Diagram, IEnumerable(String))

Initializes a new instance of the PagedContainerNode class with the specified page titles.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PagedContainerNode (
    Diagram parent,
    IEnumerable<string> pageTitles
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram, _
    pageTitles As IEnumerable(Of String) _
)
```

Parameters

parent
The parent Diagram instance.

pageTitles
A list of titles for the initial pages.

See Also

PagedContainerNode Constructor Overload List
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode Constructor (PagedContainerNode)

Initializes a new instance of the PagedContainerNode class using the specified node as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PagedContainerNode (
    PagedContainerNode prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As PagedContainerNode _
)
```

Parameters

prototype
The prototype to copy.

See Also

PagedContainerNode Constructor Overload List
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode Methods

PagedContainerNode.AddPage Method

Adds a page to the container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void AddPage (
    ContainerPage page
)
```

Visual Basic Copy Code

```
Public Sub AddPage( _
    page As ContainerPage _
)
```

Parameters

page
The page to add.

Remarks

Container's current pages are accessible via Pages collection.

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace
RemovePage Method

PagedContainerNode.Clone Method

Creates a copy of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override DiagramItem Clone (
    bool clipboard
)
```

Visual Basic Copy Code

```
Public Overrides Function Clone( _
    clipboard As Boolean _
) As DiagramItem
```

Parameters

clipboard

Return Value

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.GetCurrentItem Method

Implements InplaceEditableList.GetCurrentItem.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int GetCurrentItem ()
```

Visual Basic Copy Code

```
Public Function GetCurrentItem () As Integer
```

Return Value

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.GetListItems Method

Implements InplaceEditableList.GetListItems.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IList<string> GetListItems ()
```

Visual Basic Copy Code

```
Public Function GetListItems () As IList(Of String)
```

Return Value

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.InsertPage Method

Inserts a page at the specified index.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void InsertPage (
    int index,
    ContainerPage page
)
```

Visual Basic Copy Code

```
Public Sub InsertPage( _
    index As Integer, _
    page As ContainerPage _
)
```

Parameters

index
The zero-based index at which the page should be inserted.

page
The page to insert.

Remarks

Container's current pages are accessible via Pages collection.

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.LoadFromJson Method

Loads the item content from a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

context

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.LoadFromXml Method

Loads the item content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

context

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.MovePage Method

Moves a page to a new position in the page list.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void MovePage (
    int newIndex,
    ContainerPage page
)
```

Visual Basic Copy Code

```
Public Sub MovePage( _
    newIndex As Integer, _
    page As ContainerPage _
)
```

Parameters

newIndex
The new index of the page.

page
The page to move.

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.RemovePage Method

Removes a page from the container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RemovePage (
    ContainerPage page
)
```

Visual Basic Copy Code

```
Public Sub RemovePage( _
    page As ContainerPage _
)
```

Parameters

page
The page to remove.

Remarks

Container's current pages are accessible via Pages collection.

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace
AddPage Method

PagedContainerNode.SaveToJson Method

Saves the item content into a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

context

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.SaveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

context

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.SetCurrentItem Method

Implements InplaceEditableList.SetCurrentItem.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetCurrentItem (
    int index
)
```

Visual Basic Copy Code

```
Public Sub SetCurrentItem( _
    index As Integer _
)
```

Parameters

index

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode Properties

PagedContainerNode.CurrentPage Property

Gets or sets the current page.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ContainerPage CurrentPage { get; set; }
```

Visual Basic Copy Code

```
Public Property CurrentPage As ContainerPage
```

Property Value

A ContainerPage reference.

Remarks

PagedContainerNode displays only child items associated with this page. Users can change the current page by clicking on arrow icons in the caption bar.

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PagedContainerNode.Pages Property

Gets the pages of this container.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<ContainerPage> Pages { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Pages As List(Of ContainerPage)
```

Property Value

A list of ContainerPage objects.

Remarks

Call AddPage and RemovePage methods to modify this collection.

See Also

PagedContainerNode Members
PagedContainerNode Class
MindFusion.Diagramming Namespace

PageIconRenderer Class

Draws page navigation arrows in PagedContainerNode caption bar.

Syntax

C# Copy Code

```
public class PageIconRenderer
```

Visual Basic Copy Code

```
Public Class PageIconRenderer
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.PageIconRenderer

See Also

PageIconRenderer Members
MindFusion.Diagramming Namespace

PageIconRenderer Members

Draws page navigation arrows in PagedContainerNode caption bar.

The following tables list the members exposed by the PageIconRenderer type.

Public Constructors

	Name	Description

	PageIconRenderer	Overloaded.

Public Properties

	Name	Description
	Size	An optional double value specifying scroll arrows' size in diagram's MeasureUnit.

See Also

PageIconRenderer Class
MindFusion.Diagramming Namespace

PageIconRenderer Constructor

Overload List

Name	Description
PageIconRenderer (IImage, IImage)	Initializes a new instance of the PageIconRenderer class that renders scroll arrow icons provided as bitmaps.
PageIconRenderer (IImage, IImage, IImage, IImage)	Initializes a new instance of the **PageIconRenderer** class that renders scroll arrow icons provided as bitmaps.
PageIconRenderer (RenderCallback(PagedContainerNode))	Initializes a new instance of the **PageIconRenderer** class.
PageIconRenderer (SvgContent, SvgContent)	Initializes a new instance of the **PageIconRenderer** class that renders collapse / expand icons provided as SVG drawings.
PageIconRenderer (SvgContent, SvgContent, SvgContent, SvgContent)	Initializes a new instance of the PageIconRenderer class that renders collapse / expand icons provided as SVG drawings.

See Also

PageIconRenderer Class
MindFusion.Diagramming Namespace

PageIconRenderer Constructor (IImage, IImage)

Initializes a new instance of the PageIconRenderer class that renders scroll arrow icons provided as bitmaps.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PageIconRenderer (
    IImage scrollLeftIcon,
    IImage scrollRightIcon
)
```

Visual Basic Copy Code

```
Public New ( _
    scrollLeftIcon As IImage, _
    scrollRightIcon As IImage _
)
```

Parameters

scrollLeftIcon
A **Microsoft.Maui.Graphics.IImage** representing the scroll-left icon.

scrollRightIcon
A **Microsoft.Maui.Graphics.IImage** representing the scroll-right icon.

See Also

PageIconRenderer Constructor Overload List
PageIconRenderer Class
MindFusion.Diagramming Namespace

PageIconRenderer Constructor (IImage, IImage, IImage, IImage)

Initializes a new instance of the PageIconRenderer class that renders scroll arrow icons provided as bitmaps.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PageIconRenderer (
    IImage scrollLeftIcon,
    IImage scrollRightIcon,
    IImage disabledScrollLeftIcon,
    IImage disabledScrollRightIcon
)
```

Visual Basic Copy Code

```
Public New ( _
    scrollLeftIcon As IImage, _
    scrollRightIcon As IImage, _
    disabledScrollLeftIcon As IImage, _
    disabledScrollRightIcon As IImage _
)
```

Parameters

scrollLeftIcon
A **Microsoft.Maui.Graphics.IImage** representing the scroll-left icon.

scrollRightIcon
A **Microsoft.Maui.Graphics.IImage** representing the scroll-right icon.

disabledScrollLeftIcon
A **Microsoft.Maui.Graphics.IImage** representing disabled scroll-left icon.

disabledScrollRightIcon
A **Microsoft.Maui.Graphics.IImage** representing disabled scroll-right icon.

See Also

PageIconRenderer Constructor Overload List
PageIconRenderer Class
MindFusion.Diagramming Namespace

PageIconRenderer Constructor (RenderCallback(PagedContainerNode))

Initializes a new instance of the PageIconRenderer class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PageIconRenderer (
    RenderCallback<PagedContainerNode> renderCallback
)
```

Visual Basic Copy Code

```
Public New ( _
    renderCallback As RenderCallback(Of PagedContainerNode) _
)
```

Parameters

renderCallback

A callback function that draws the scroll arrows using IGraphics code.

See Also

PageIconRenderer Constructor Overload List
PageIconRenderer Class
MindFusion.Diagramming Namespace

PageIconRenderer Constructor (SvgContent, SvgContent)

Initializes a new instance of the PageIconRenderer class that renders collapse / expand icons provided as SVG drawings.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PageIconRenderer (
    SvgContent scrollLeftIcon,
    SvgContent scrollRightIcon
)
```

Visual Basic Copy Code

```
Public New ( _
    scrollLeftIcon As SvgContent, _
    scrollRightIcon As SvgContent _
)
```

Parameters

scrollLeftIcon

An SvgContent representing the scroll-left icon.

scrollRightIcon

An SvgContent representing the scroll-right icon.

See Also

PageIconRenderer Constructor Overload List
PageIconRenderer Class
MindFusion.Diagramming Namespace

PageIconRenderer Constructor (SvgContent, SvgContent, SvgContent, SvgContent)

Initializes a new instance of the PageIconRenderer class that renders collapse / expand icons provided as SVG drawings.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PageIconRenderer (
    SvgContent scrollLeftIcon,
    SvgContent scrollRightIcon,
    SvgContent disabledScrollLeftIcon,
    SvgContent disabledScrollRightIcon
)
```

Visual Basic Copy Code

```
Public New ( _
    scrollLeftIcon As SvgContent, _
    scrollRightIcon As SvgContent, _
    disabledScrollLeftIcon As SvgContent, _
    disabledScrollRightIcon As SvgContent _
)
```

Parameters

scrollLeftIcon
An SvgContent representing the scroll-left icon.

scrollRightIcon
An SvgContent representing the scroll-right icon.

disabledScrollLeftIcon
An SvgContent representing disabled scroll-left icon.

disabledScrollRightIcon
An SvgContent representing disabled scroll-right icon.

See Also

PageIconRenderer Constructor Overload List
PageIconRenderer Class
MindFusion.Diagramming Namespace

PageIconRenderer Properties

PageIconRenderer.Size Property

An optional double value specifying scroll arrows' size in diagram's MeasureUnit.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Nullable<Double> Size { get; set; }
```

Visual Basic Copy Code

```
Public Property Size As Nullable(Of Double)
```

Property Value

See Also

PageIconRenderer Members
PageIconRenderer Class

Path Class

Represents a sequence of connected DiagramNode and DiagramLink elements.

Syntax

C# Copy Code

```
public sealed class Path
```

Visual Basic Copy Code

```
Public NotInheritable Class Path
```

Remarks

Path instances are returned by methods of the PathFinder class.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Path

See Also

Path Members
MindFusion.Diagramming Namespace

Path Members

Represents a sequence of connected DiagramNode and DiagramLink elements.

The following tables list the members exposed by the Path type.

Public Properties

	Name	Description
	Items	Gets a collection of all DiagramNode and DiagramLink elements in the path.
	Length	Gets the length of the **Path** expressed as the number of Links participating in it.
	Links	Gets a collection of all DiagramLink elements participating in the path.
	Nodes	Gets a collection of all DiagramNode elements participating in the path.

Public Methods

	Name	Description
	GetWeight	Calculates the weight of this **Path** by accumulating the Weight of all participating Diagram elements.

See Also

Path Class
MindFusion.Diagramming Namespace

Path Methods

Path.GetWeight Method

Calculates the weight of this Path by accumulating the Weight of all participating Diagram elements.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public float GetWeight (
    bool incNodes,
    bool incLinks
)
```

Visual Basic Copy Code

```
Public Function GetWeight( _
    incNodes As Boolean, _
    incLinks As Boolean _
) As Single
```

Parameters

incNodes
A boolean value, which specifies whether the Weight of DiagramNode elements should be considered.

incLinks
A boolean value, which specifies whether the **Weight** of DiagramLink elements should be considered.

Return Value

A float value that represents the total weight of the elements in this Path according to the *incNodes* and *incLinks* parameters.

See Also

Path Members
Path Class
MindFusion.Diagramming Namespace

Path Properties

Path.Items Property

Gets a collection of all DiagramNode and DiagramLink elements in the path.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItemCollection Items { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Items As DiagramItemCollection
```

Property Value

An instance of the DiagramItemCollection class.

Remarks

The collection contains consecutive DiagramNode and DiagramLink objects.

See Also

Path Members
Path Class
MindFusion.Diagramming Namespace

Path.Length Property

Gets the length of the Path expressed as the number of Links participating in it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Length { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Length As Integer
```

Property Value

An integer value that represents the number of edges (links) in the path.

See Also

Path Members
Path Class
MindFusion.Diagramming Namespace

Path.Links Property

Gets a collection of all DiagramLink elements participating in the path.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection Links { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Links As DiagramLinkCollection
```

Property Value

An instance of the DiagramLinkCollection class.

Remarks

Use Nodes to get all DiagramNode elements in the Path.

See Also

Path Members
Path Class
MindFusion.Diagramming Namespace

Path.Nodes Property

Gets a collection of all DiagramNode elements participating in the path.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeCollection Nodes { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Nodes As DiagramNodeCollection
```

Property Value

An instance of the DiagramNodeCollection class.

Remarks

Use Links to get all DiagramLinks elements in the Path.

See Also

Path Members
Path Class
MindFusion.Diagramming Namespace

PathCollection Class

Represents a strongly typed collection of Path objects.

Syntax

C# Copy Code

```
public sealed class PathCollection : CollectionBase
```

Visual Basic Copy Code

```
Public NotInheritable Class PathCollection
    Inherits CollectionBase
```

Inheritance Hierarchy

System.Object
  System.Collections.CollectionBase
    MindFusion.Diagramming.PathCollection

See Also

PathCollection Members
MindFusion.Diagramming Namespace

PathCollection Members

Represents a strongly typed collection of Path objects.

The following tables list the members exposed by the PathCollection type.

Public Constructors

	Name	Description
	PathCollection	Initializes a new instance of the **PathCollection** class.

Public Properties

	Name	Description
	Item	Gets or sets the Path at the specified position.

Public Methods

	Name	Description
	Add	Adds the specified Path to the collection.
	Clear	Removes all Path elements from the **PathCollection**.
	GetAt	Gets the Path at the specified position.
	Insert	Inserts the specified Path to the **PathCollection** at the specified index.
	Remove	Removes the specified Path element from the **PathCollection**.

See Also

PathCollection Class
MindFusion.Diagramming Namespace

PathCollection Constructor

Initializes a new instance of the PathCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PathCollection ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

PathCollection Class
MindFusion.Diagramming Namespace

PathCollection Methods

PathCollection.Add Method

Adds the specified Path to the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

Copy Code

```
public void Add (
    Path path
)
```

Visual Basic Copy Code

```
Public Sub Add( _
    path As Path _
)
```

Parameters

path

The Path object to add.

Remarks

The Path object is added at the end of the PathCollection. Use Insert to specify the place of the new element.

See Also

PathCollection Members
PathCollection Class
MindFusion.Diagramming Namespace

PathCollection.Clear Method

Removes all Path elements from the PathCollection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public new void Clear ()
```

Visual Basic Copy Code

```
Public Shadows Sub Clear ()
```

Remarks

The method removes all Path elements in the collection. If you want to delete a specific element use Remove.

See Also

PathCollection Members
PathCollection Class
MindFusion.Diagramming Namespace

PathCollection.GetAt Method

Gets the Path at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Path GetAt (
    int index
```

```
)
```

Visual Basic Copy Code

```
Public Function GetAt( _
    index As Integer _
) As Path
```

Parameters

index

The zero-based index of the Path element to get.

Return Value

The **Path** at the specified position in the PathCollection.

See Also

PathCollection Members
PathCollection Class
MindFusion.Diagramming Namespace

PathCollection.Insert Method

Inserts the specified Path to the PathCollection at the specified index.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Insert (
    int index,
    Path path
)
```

Visual Basic Copy Code

```
Public Sub Insert( _
    index As Integer, _
    path As Path _
)
```

Parameters

index

The zero-based index at which the new Path element should be inserted.

path

The **Path** element to insert.

Remarks

The method inserts a new element at a given position in the PathCollection. Use Add to add a new element to the end of the collection.

See Also

PathCollection Members
PathCollection Class
MindFusion.Diagramming Namespace

PathCollection.Remove Method

Removes the specified Path element from the PathCollection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Remove (
    Path path
)
```

Visual Basic Copy Code

```
Public Sub Remove( _
    path As Path _
)
```

Parameters

path
The Path element to remove.

Remarks

The size of the PathCollection is corrected. Indexes of the elements after *path* are updated too.

See Also

PathCollection Members
PathCollection Class
MindFusion.Diagramming Namespace

PathCollection Properties

PathCollection.Item Property

Gets or sets the Path at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Path Item [
    int index
] { get; set; }
```

Visual Basic Copy Code

```
Public Property Item( _
    index As Integer _
) As Path
```

Parameters

index
The zero-based index of the Path element to access.

Property Value

An instance of the **Path** class that represents the path at the specified position.

See Also

PathFinder Class

Provides methods that help you find paths and cycles in a graph.

Syntax

C# Copy Code

```
public sealed class PathFinder
```

Visual Basic Copy Code

```
Public NotInheritable Class PathFinder
```

Remarks

By default the direction of links is taken into account when looking for paths or cycles. In that case the direction of a path is considered to be the direction defined by the outgoing links of nodes. Use the second constructor override to specify that link direction should be ignored when looking for paths.

A path-finder object caches graph information between method calls in order to achieve better performance when path and cycle -finding methods are called on the same diagram more than once. However if the diagram structure changes, you must create a new **PathFinder** instance in order to use the new structure of the diagram graph.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.PathFinder

See Also

PathFinder Members
MindFusion.Diagramming Namespace

PathFinder Members

Provides methods that help you find paths and cycles in a graph.

The following tables list the members exposed by the PathFinder type.

Public Constructors

	Name	Description
	PathFinder	Overloaded. Initializes a new instance of the **PathFinder** class.

Public Methods

	Name	Description
	EnumAllCycles	Enumerates all cycles in the underlying diagram.
	EnumAllPaths	Overloaded.
	FindAllCycles	Finds all cycles in the underlying diagram.
	FindAllPaths	Overloaded.
	FindCycle	Overloaded. Finds a cycle in the underlying diagram.
	FindLongestPath	Overloaded. Finds the longest path in the underlying diagram.

	FindShortestPath	Overloaded. Finds the shortest path in the underlying diagram.

See Also

PathFinder Class
MindFusion.Diagramming Namespace

PathFinder Constructor

Initializes a new instance of the PathFinder class.

Overload List

Name	Description
PathFinder (Diagram)	Initializes a new instance of the **PathFinder** class. Path-finding will depend on the direction of links.
PathFinder (Diagram, Boolean)	Initializes a new instance of the PathFinder class, optionally ignoring the link direction.

See Also

PathFinder Class
MindFusion.Diagramming Namespace

PathFinder Constructor (Diagram)

Initializes a new instance of the PathFinder class. Path-finding will depend on the direction of links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PathFinder (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram _
)
```

Parameters

diagram
An instance of the Diagram class in which to look for paths or cycles.

Remarks

The PathFinder methods look for paths and cycles in the Diagram instance specified through the argument. The diagram is treated as a directed graph and the direction of links is considered when looking for a path, that is, a path continues from a node only through its **OutgoingLinks**.

See Also

PathFinder Constructor Overload List
PathFinder Class
MindFusion.Diagramming Namespace

PathFinder Constructor (Diagram, Boolean)

Initializes a new instance of the PathFinder class, optionally ignoring the link direction.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PathFinder (
    Diagram diagram,
    bool ignoreDirection
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram, _
    ignoreDirection As Boolean _
)
```

Parameters

diagram
A Diagram instance in which to look for paths or cycles.

ignoreDirection
Specifies whether the direction of links should be considered when looking for paths or cycles.

Remarks

The PathFinder methods look for paths and cycles in the Diagram instance specified with the first argument. By default, the diagram is treated as a directed graph and the direction of links is considered when looking for a path, that is, a path continues from a node only through its **OutgoingLinks**.

To ignore the direction of links, pass **true** as a value of the *ignoreDirection* argument; then the path-finding algorithm will look for paths both through the **OutgoingLinks** and **IncomingLinks** collections of nodes. If you do not want to ignore the direction of links you can use the alternative constructor.

See Also

PathFinder Constructor Overload List
PathFinder Class
MindFusion.Diagramming Namespace

PathFinder Methods

PathFinder.EnumAllCycles Method

Enumerates all cycles in the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IEnumerable<Path> EnumAllCycles ()
```

Visual Basic Copy Code

```
Public Function EnumAllCycles () As IEnumerable(Of Path)
```

Return Value

A collection of Path objects; if no cycles are detected, the collection is empty.

Remarks

The **Enum*** methods of PathFinder immediately yield each path once found, as opposed to **Find*** methods which collect them in a list.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace

PathFinder.EnumAllPaths Method

Overload List

Name	Description
PathFinder.EnumAllPaths (DiagramNode, DiagramNode)	Enumerates all paths that exist between two DiagramNode objects.
PathFinder.EnumAllPaths (DiagramNode, DiagramNode, Int32)	Enumerates all paths that exist between two **DiagramNode** objects.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace

PathFinder.EnumAllPaths Method (DiagramNode, DiagramNode)

Enumerates all paths that exist between two DiagramNode objects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IEnumerable<Path> EnumAllPaths (
    DiagramNode from,
    DiagramNode to
)
```

Visual Basic Copy Code

```
Public Function EnumAllPaths( _
    from As DiagramNode, _
    to As DiagramNode _
) As IEnumerable(Of Path)
```

Parameters

from
The first DiagramNode in the path.

to
The last **DiagramNode** in the path.

Return Value

A collection of Path objects. If no paths are found, the collection is empty.

Remarks

The returned Path instances include the from and to nodes.

The **Enum*** methods of PathFinder immediately yield each path once found, as opposed to **Find*** methods which collect them in a list.

See Also

EnumAllPaths Method Overload List
PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace

PathFinder.EnumAllPaths Method (DiagramNode, DiagramNode, Int32)

Enumerates all paths that exist between two DiagramNode objects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IEnumerable<Path> EnumAllPaths (
    DiagramNode from,
    DiagramNode to,
    int maxPaths
)
```

Visual Basic Copy Code

```
Public Function EnumAllPaths( _
    from As DiagramNode, _
    to As DiagramNode, _
    maxPaths As Integer _
) As IEnumerable(Of Path)
```

Parameters

from
The first DiagramNode in the Path.

to
The last **DiagramNode** in the Path.

maxPaths
The maximum number of paths to find.

Return Value

A collection of Path objects; if no paths are found, the collection is empty.

Remarks

The returned Path instances include the from and to nodes.

The **Enum*** methods of PathFinder immediately yield each path once found, as opposed to **Find*** methods which collect them in a list.

See Also

EnumAllPaths Method Overload List
PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace

PathFinder.FindAllCycles Method

Finds all cycles in the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PathCollection FindAllCycles ()
```

Visual Basic Copy Code

```
Public Function FindAllCycles () As PathCollection
```

Return Value

A collection of Path objects; if no cycles are detected, the collection is empty.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
FindCycle Method
EnumAllCycles Method

PathFinder.FindAllPaths Method

Overload List

Name	Description
PathFinder.FindAllPaths (DiagramNode, DiagramNode)	Finds and returns all paths that exist between two DiagramNode objects.
PathFinder.FindAllPaths (DiagramNode, DiagramNode, Int32)	Finds and returns all paths that exist between two DiagramNode objects.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace

PathFinder.FindAllPaths Method (DiagramNode, DiagramNode)

Finds and returns all paths that exist between two DiagramNode objects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PathCollection FindAllPaths (
    DiagramNode from,
    DiagramNode to
)
```

Visual Basic Copy Code

```
Public Function FindAllPaths( _
    from As DiagramNode, _
    to As DiagramNode _
) As PathCollection
```

Parameters

from
The first DiagramNode in the Path.

to
The last **DiagramNode** in the **Path**.

Return Value

A collection of **Path** objects; if no paths are found, the collection is empty.

Remarks

The returned Path instances include the *from* an *to* nodes.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
FindShortestPath Method
EnumAllPaths Method

PathFinder.FindAllPaths Method (DiagramNode, DiagramNode, Int32)

Finds and returns all paths that exist between two DiagramNode objects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PathCollection FindAllPaths (
    DiagramNode from,
    DiagramNode to,
    int maxPaths
)
```

Visual Basic Copy Code

```
Public Function FindAllPaths( _
    from As DiagramNode, _
    to As DiagramNode, _
    maxPaths As Integer _
) As PathCollection
```

Parameters

from
The first DiagramNode in the Path.

to
The last DiagramNode in the Path.

maxPaths
The maximum number of paths to find.

Return Value

A collection of Path objects; if no paths are found, the collection is empty.

Remarks

The returned Path instances include the from and to nodes.

See Also

FindAllPaths Method Overload List
PathFinder Members
PathFinder Class

PathFinder.FindCycle Method

Finds a cycle in the underlying diagram.

Overload List

Name	Description
PathFinder.FindCycle ()	Finds a cycle in the Diagram.
PathFinder.FindCycle (DiagramNode)	Detects whether the specified DiagramNode participates in a cycle.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace

PathFinder.FindCycle Method ()

Finds a cycle in the Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Path FindCycle ()
```

Visual Basic Copy Code

```
Public Function FindCycle () As Path
```

Return Value

A Path instance that represents the found cycle; **null** (**Nothing** in Visual Basic) if no cycle is detected.

See Also

FindCycle Method Overload List
PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
FindShortestPath Method

PathFinder.FindCycle Method (DiagramNode)

Detects whether the specified DiagramNode participates in a cycle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Path FindCycle (
    DiagramNode participant
)
```

Visual Basic

```
Public Function FindCycle( _
    participant As DiagramNode _
) As Path
```

Parameters

participant
A DiagramNode element which should participate in the found cycle.

Return Value

A Path instance that represents the found cycle; **null** (**Nothing** in Visual Basic) if no cycle is detected.

See Also

FindCycle Method Overload List
PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
FindAllCycles Method

PathFinder.FindLongestPath Method

Finds the longest path in the underlying diagram.

Overload List

Name	Description
PathFinder.FindLongestPath ()	Finds the longest path in the graph.
PathFinder.FindLongestPath (DiagramNode, DiagramNode)	Finds the longest path between the specified DiagramNode objects.
PathFinder.FindLongestPath (Int64)	Finds the longest path in the diagram.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace

PathFinder.FindLongestPath Method ()

Finds the longest path in the graph.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Path FindLongestPath ()
```

Visual Basic

```
Public Function FindLongestPath () As Path
```

Return Value

A Path instance that represents the longest path found; **null** (**Nothing** in Visual Basic) if there are no DiagramNode elements in the graph.

See Also

FindLongestPath Method Overload List
PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
FindAllPaths Method

PathFinder.FindLongestPath Method (DiagramNode, DiagramNode)

Finds the longest path between the specified DiagramNode objects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Path FindLongestPath (
    DiagramNode from,
    DiagramNode to
)
```

Visual Basic Copy Code

```
Public Function FindLongestPath( _
    from As DiagramNode, _
    to As DiagramNode _
) As Path
```

Parameters

from
The first DiagramNode element in the Path.

to
The last **DiagramNode** element in the **Path**.

Return Value

A **Path** instance that represents the longest path found; **null** (**Nothing** in Visual Basic) if there is no **Path** connecting *from* and *to*.

See Also

FindLongestPath Method Overload List
PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
FindAllPaths Method

PathFinder.FindLongestPath Method (Int64)

Finds the longest path in the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Path FindLongestPath (
    long timeLimit
)
```

Visual Basic Copy Code

```
Public Function FindLongestPath( _
    timeLimit As Long _
```

```
) As Path
```

Parameters

timeLimit
The maximum duration of the search process, in milliseconds.

Return Value

The longest Path found; **null** (**Nothing** in Visual Basic) if no **Path** has been detected.

See Also

FindLongestPath Method Overload List
PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
FindAllPaths Method

PathFinder.FindShortestPath Method

Finds the shortest path in the underlying diagram.

Overload List

Name	Description
PathFinder.FindShortestPath (DiagramNode, DiagramNode)	Finds and returns the shortest Path between the specified DiagramNode objects.
PathFinder.FindShortestPath (DiagramNode, DiagramNode, Boolean, Boolean)	Finds and returns the shortest path between two DiagramNode elements, considering the Weight of the nodes, the **Weight** of the links or both.

See Also

PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace

PathFinder.FindShortestPath Method (DiagramNode, DiagramNode)

Finds and returns the shortest Path between the specified DiagramNode objects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Path FindShortestPath (
    DiagramNode from,
    DiagramNode to
)
```

Visual Basic Copy Code

```
Public Function FindShortestPath( _
    from As DiagramNode, _
    to As DiagramNode _
) As Path
```

Parameters

from
The first DiagramNode in the Path.

to

The last **DiagramNode** in the **Path**.

Return Value

A **Path** instance that represents the shortest path found; **null** (**Nothing** in Visual Basic) if there is no path connecting *from* and *to*.

See Also

FindShortestPath Method Overload List
PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
FindAllPaths Method

PathFinder.FindShortestPath Method (DiagramNode, DiagramNode, Boolean, Boolean)

Finds and returns the shortest path between two DiagramNode elements, considering the Weight of the nodes, the **Weight** of the links or both.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Path FindShortestPath (
    DiagramNode from,
    DiagramNode to,
    bool useNodeWeights,
    bool useLinkWeights
)
```

Visual Basic Copy Code

```
Public Function FindShortestPath( _
    from As DiagramNode, _
    to As DiagramNode, _
    useNodeWeights As Boolean, _
    useLinkWeights As Boolean _
) As Path
```

Parameters

from

The first DiagramNode in the Path.

to

The last **DiagramNode** in the **Path**.

useNodeWeights

If **true**, the **Path** with minimal total weight is found, summing the Weight of nodes in the **Path**.

useLinkWeights

If **true**, the **Path** with minimal total weight is found, summing the **Weight** of links in the **Path**.

Return Value

A **Path** instance that represents the shortest **Path** found or **null** (**Nothing** in Visual Basic) if there is no **Path** connecting *from* and *to*.

See Also

FindShortestPath Method Overload List
PathFinder Members
PathFinder Class
MindFusion.Diagramming Namespace
FindAllPaths Method

PatternRouter Class

Routes links by applying patterns of segment and turn sequences and accepting them when resulting paths do not cross nodes.

Syntax

C# Copy Code

```
public class PatternRouter : ILinkRouter
```

Visual Basic Copy Code

```
Public Class PatternRouter
    Implements ILinkRouter
```

Remarks

You can add custom patterns to the Patterns list, or replace the standard patterns altogether. Patterns are defined using RoutePattern and RouteStep objects.

Example

The following code defines a Z-like link route that will be accepted if it does not cross Obstacle nodes, nor passes by them within MinimumDistance.

C# Copy Code

```
var rightZigzag = new RoutePattern();
rightZigzag.Steps.Add(
    new RouteStep(RouteStepKind.AdvanceRelative, 0.5f));
rightZigzag.Steps.Add(
    new RouteStep(RouteStepKind.TurnRight));
rightZigzag.Steps.Add(
    new RouteStep(RouteStepKind.AdvanceRelative, 1));
rightZigzag.Steps.Add(
    new RouteStep(RouteStepKind.TurnLeft));
rightZigzag.Steps.Add(
    new RouteStep(RouteStepKind.AdvanceRelative, 0.5f));
router.Patterns.Add(rightZigzag);

// this can be defined in shorter form using a pattern string
// var rightZigzag = new RoutePattern(
//     "ADR0.5 TRR ADR1 TRL ADR0.5");
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.PatternRouter

See Also

PatternRouter Members
MindFusion.Diagramming Namespace

PatternRouter Members

Routes links by applying patterns of segment and turn sequences and accepting them when resulting paths do not cross nodes.

The following tables list the members exposed by the PatternRouter type.

Public Constructors

	Name	Description
	PatternRouter	Overloaded.

Public Properties

	Name	Description
	Diagram	Gets or sets the Diagram whose links will be routed.
	MinimumDistance	Gets or sets the minimum allowed distance from obstacles for a route to be accepted.
	Obstacles	Gets or sets a list of obstacle nodes.
	Patterns	Gets or sets a list of RoutePattern objects representing the routing patterns.
	PreferredDistance	Gets or sets the preferred distance from link's bend points to the link's nodes.

Public Methods

	Name	Description
	RouteLink	Implements ILinkRouter.RouteLink.
	RouteLinks	Implements ILinkRouter.RouteLinks.

See Also

PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter Constructor

Overload List

Name	Description
PatternRouter ()	Initializes a new instance of the PatternRouter class.
PatternRouter (Diagram)	Initializes a new instance of the PatternRouter class.
PatternRouter (Diagram, List(RoutePattern))	Initializes a new instance of the PatternRouter class.

See Also

PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter Constructor ()

Initializes a new instance of the PatternRouter class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PatternRouter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

PatternRouter Constructor Overload List
PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter Constructor (Diagram)

Initializes a new instance of the PatternRouter class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PatternRouter (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram _
)
```

Parameters

diagram
The Diagram whose links will be routed.

See Also

PatternRouter Constructor Overload List
PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter Constructor (Diagram, List(RoutePattern))

Initializes a new instance of the PatternRouter class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PatternRouter (
    Diagram diagram,
    List<RoutePattern> patterns
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram, _
    patterns As List(Of RoutePattern) _
)
```

Parameters

diagram
The Diagram whose links will be routed.

patterns
A list of RoutePattern objects.

See Also

PatternRouter Constructor Overload List
PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter Methods

PatternRouter.RouteLink Method

Implements ILinkRouter.RouteLink.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RouteLink (
    DiagramLink link,
    ConnectionPoint originConnection,
    ConnectionPoint destinationConnection,
    bool nowCreating
)
```

Visual Basic Copy Code

```
Public Function RouteLink( _
    link As DiagramLink, _
    originConnection As ConnectionPoint, _
    destinationConnection As ConnectionPoint, _
    nowCreating As Boolean _
) As Boolean
```

Parameters

link
The DiagramLink to route.

originConnection
Represents connection to Origin node, potentially different from link's stored connection when modifying first control point.

destinationConnection
Represents connection to Destination node, potentially different from link's stored connection when modifying last control point.

nowCreating
Indicates whether link is currently being created.

Return Value

true if a path has been found successfully, otherwise false.

See Also

PatternRouter Members
PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter.RouteLinks Method

Implements ILinkRouter.RouteLinks.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection RouteLinks (
    DiagramLinkCollection links
)
```

Visual Basic Copy Code

```
Public Function RouteLinks( _
    links As DiagramLinkCollection _
) As DiagramLinkCollection
```

Parameters

links

A list of DiagramLink objects that should be routed.

Return Value

A list of links that were not routed successfully.

See Also

PatternRouter Members
PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter Properties

PatternRouter.Diagram Property

Gets or sets the Diagram whose links will be routed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Diagram Diagram { get; set; }
```

Visual Basic Copy Code

```
Public Property Diagram As Diagram
```

Property Value

An instance of the Diagram class.

See Also

PatternRouter Members
PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter.MinimumDistance Property

Gets or sets the minimum allowed distance from obstacles for a route to be accepted.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double MinimumDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property MinimumDistance As Double
```

Property Value

A **double** value specifying minimum distance. The default value is 4 millimeters.

See Also

PatternRouter Members
PatternRouter Class
MindFusion.Diagramming Namespace
PreferredDistance Property

PatternRouter.Obstacles Property

Gets or sets a list of obstacle nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeCollection Obstacles { get; set; }
```

Visual Basic Copy Code

```
Public Property Obstacles As DiagramNodeCollection
```

Property Value

A list of obstacles.

Remarks

If this value is set, the router ignores diagram's Nodes and only considers nodes from specified list.

See Also

PatternRouter Members
PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter.Patterns Property

Gets or sets a list of RoutePattern objects representing the routing patterns.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<RoutePattern> Patterns { get; set; }
```

Visual Basic Copy Code

```
Public Property Patterns As List(Of RoutePattern)
```

Property Value

A list of RoutePattern objects defining link route templates.

See Also

PatternRouter Members
PatternRouter Class
MindFusion.Diagramming Namespace

PatternRouter.PreferredDistance Property

Gets or sets the preferred distance from link's bend points to the link's nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double PreferredDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property PreferredDistance As Double
```

Property Value

A **double** value specifying preferred distance.

Remarks

The value of this property can be used in route templates using an AdvancePreferred RouteStep.

See Also

PatternRouter Members
PatternRouter Class
MindFusion.Diagramming Namespace
MinimumDistance Property

PointCollection Class

Represents a strongly typed collection of points.

Syntax

C# Copy Code

```
public class PointCollection : List<Point>
```

Visual Basic Copy Code

```
Public Class PointCollection
    Inherits List(Of Point)
```

Inheritance Hierarchy

System.Object
  System.Collections.Generic.List(Point)
    MindFusion.Diagramming.PointCollection

See Also

PointCollection Members

Represents a strongly typed collection of points.

The following tables list the members exposed by the PointCollection type.

Public Constructors

	Name	Description
	PointCollection	Overloaded. Initializes a new instance of the **PointCollection** class.

Public Properties

	Name	Description
	Item	Gets or sets the point at the specified position.

Public Methods

	Name	Description
	Add	Adds the specified point to the collection.
	AddRange	Adds the points in the specified collection to this collection.
	Clear	Removes all points from the collection.
	Clone	Creates a shallow copy of this collection.
	GetArray	Gets a **Point** array containing the points in this collection.
	GetAt	Gets the point at the specified position.
	Insert	Inserts the specified point in the collection at the specified position.
	Remove	Removes the specified point from the collection.
	Reverse	Returns a copy of this collection with reversed order of elements.
	SetAt	Sets the specified point at the specified position within the collection.

See Also

PointCollection Class
MindFusion.Diagramming Namespace

PointCollection Constructor

Initializes a new instance of the PointCollection class.

Overload List

Name	Description
PointCollection (IList(Point))	Initializes a new instance of the PointCollection class with the specified point values.
PointCollection (Int32)	Initializes a new instance of the **PointCollection** class with the specified initial size.

PointCollection (Point[])	Initializes a new instance of the **PointCollection** class with the specified point values.

See Also

PointCollection Class
MindFusion.Diagramming Namespace

PointCollection Constructor (IList(Point))

Initializes a new instance of the PointCollection class with the specified point values.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PointCollection (
    IList<Point> points
)
```

Visual Basic Copy Code

```
Public New ( _
    points As IList(Of Point) _
)
```

Parameters

points
An array of Point values that supplies the elements initially contained in the collection.

See Also

PointCollection Constructor Overload List
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection Constructor (Int32)

Initializes a new instance of the PointCollection class with the specified initial size.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PointCollection (
    int size
)
```

Visual Basic Copy Code

```
Public New ( _
    size As Integer _
)
```

Parameters

size
The initial size of the collection.

See Also

PointCollection Constructor Overload List
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection Constructor (Point[])

Initializes a new instance of the PointCollection class with the specified point values.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PointCollection (
    Point[] points
)
```

Visual Basic Copy Code

```
Public New ( _
    points() As Point _
)
```

Parameters

points
An array of **Microsoft.Maui.Graphics.Point** values that supplies the elements initially contained in the collection.

See Also

PointCollection Constructor Overload List
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection Methods

PointCollection.Add Method

Adds the specified point to the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Add (
    Point point
)
```

Visual Basic Copy Code

```
Public Sub Add( _
    point As Point _
)
```

Parameters

point
The **Microsoft.Maui.Graphics.Point** to add to the collection.

See Also

PointCollection Members
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection.AddRange Method

Adds the points in the specified collection to this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void AddRange (
    ICollection c
)
```

Visual Basic Copy Code

```
Public Sub AddRange( _
    c As ICollection _
)
```

Parameters

c
The .NET **ICollection** object that contains the points to add to this collection.

See Also

PointCollection Members
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection.Clear Method

Removes all points from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public new void Clear ()
```

Visual Basic Copy Code

```
Public Shadows Sub Clear ()
```

See Also

PointCollection Members
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection.Clone Method

Creates a shallow copy of this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PointCollection Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As PointCollection
```

Return Value

An instance of the PointCollection class.

See Also

PointCollection Members
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection.GetArray Method

Gets a **Point** array containing the points in this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point[] GetArray ()
```

Visual Basic Copy Code

```
Public Function GetArray () As Point()
```

Return Value

An array of **Microsoft.Maui.Graphics.Point** objects containing all points in the collection.

See Also

PointCollection Members
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection.GetAt Method

Gets the point at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point GetAt (
    int index
)
```

Visual Basic Copy Code

```
Public Function GetAt( _
    index As Integer _
) As Point
```

Parameters

index
The zero-based index of the point to retrieve.

Return Value

The **Point** object at the specified index.

See Also

PointCollection Members
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection.Insert Method

Inserts the specified point in the collection at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Insert (
    int index,
    Point point
)
```

Visual Basic Copy Code

```
Public Sub Insert( _
    index As Integer, _
    point As Point _
)
```

Parameters

index
The zero-based index at which to insert the point.

point
The **Microsoft.Maui.Graphics.Point** to add to the collection.

See Also

PointCollection Members
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection.Remove Method

Removes the specified point from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Remove (
    Point point
```

```
)
```

Visual Basic Copy Code

```
Public Sub Remove( _
    point As Point _
)
```

Parameters

point
The **Microsoft.Maui.Graphics.Point** to remove.

See Also

PointCollection Members
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection.Reverse Method

Returns a copy of this collection with reversed order of elements.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public new PointCollection Reverse ()
```

Visual Basic Copy Code

```
Public Shadows Function Reverse () As PointCollection
```

Return Value

The reverse collection.

See Also

PointCollection Members
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection.SetAt Method

Sets the specified point at the specified position within the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetAt (
    int index,
    Point point
)
```

Visual Basic Copy Code

```
Public Sub SetAt( _
    index As Integer, _
```

```
    point As Point _
)
```

Parameters

index
The zero-based index of the point to set.

point
The **Microsoft.Maui.Graphics.Point** to set at the specified position.

See Also

PointCollection Members
PointCollection Class
MindFusion.Diagramming Namespace

PointCollection Properties

PointCollection.Item Property

Gets or sets the point at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point Item [
    int index
] { get; set; }
```

Visual Basic Copy Code

```
Public Property Item( _
    index As Integer _
) As Point
```

Parameters

index
The zero-based index of the point to set or retrieve.

Property Value

The **Microsoft.Maui.Graphics.Point** object at the specified index.

See Also

PointCollection Members
PointCollection Class
MindFusion.Diagramming Namespace

PolylineRouting Enumeration

Defines values that indicate how Polyline links should be routed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum PolylineRouting
```

Visual Basic	Copy Code
`Public Enum PolylineRouting`	

Members

	Member name	Description
	UnrestrictedInclined	Allow any segment to be inclined by non-multiple of 90 angles.
	EndSegmentsOrthogonal	Keep the end segments orthogonal to node's sides, and allow interior segments to be inclined.
	AllOrthogonal	Keep all segments orthogonal to node's sides and each other.

See Also

MindFusion.Diagramming Namespace

QuickRouter Class

Implements a very fast link routing algorithm that uses a tree to represent the free and occupied regions in the diagram.

Syntax

C#	Copy Code
`public class QuickRouter : ILinkRouter`	

Visual Basic	Copy Code
`Public Class QuickRouter` `    Implements ILinkRouter`	

Remarks

To use this class for link routing simply assign its instance to the LinkRouter property of the Diagram class.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.QuickRouter

See Also

QuickRouter Members
MindFusion.Diagramming Namespace

QuickRouter Members

Implements a very fast link routing algorithm that uses a tree to represent the free and occupied regions in the diagram.

The following tables list the members exposed by the QuickRouter type.

Public Constructors

	Name	Description
	QuickRouter	Overloaded.

Public Properties

	Name	Description

	Diagram	Gets or sets the diagram this router is associated with.
	Granularity	Gets or sets the granularity of the obstacles map generated by **QuickRouter**.
	TurnCost	Gets or sets a value added to the total cost of a route if it makes a turn.
	UBendMaxLen	Gets or sets the maximum U-bend length.

Public Methods

	Name	Description
	RouteAllLinks	Routes all links in the diagram.
	RouteLink	Implements ILinkRouter.RouteLink.
	RouteLinks	Implements ILinkRouter.RouteLinks.

See Also

QuickRouter Class
MindFusion.Diagramming Namespace

QuickRouter Constructor

Overload List

Name	Description
QuickRouter ()	Initializes a new instance of the QuickRouter class.
QuickRouter (Diagram)	Initializes a new instance of the QuickRouter class.

See Also

QuickRouter Class
MindFusion.Diagramming Namespace

QuickRouter Constructor ()

Initializes a new instance of the QuickRouter class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public QuickRouter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

QuickRouter Constructor Overload List
QuickRouter Class
MindFusion.Diagramming Namespace

QuickRouter Constructor (Diagram)

Initializes a new instance of the QuickRouter class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public QuickRouter (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram _
)
```

Parameters

diagram
The Diagram this router will operate on.

See Also

QuickRouter Constructor Overload List
QuickRouter Class
MindFusion.Diagramming Namespace

QuickRouter Methods

QuickRouter.RouteAllLinks Method

Routes all links in the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RouteAllLinks ()
```

Visual Basic Copy Code

```
Public Sub RouteAllLinks ()
```

See Also

QuickRouter Members
QuickRouter Class
MindFusion.Diagramming Namespace

QuickRouter.RouteLink Method

Implements ILinkRouter.RouteLink.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RouteLink (
    DiagramLink link,
    ConnectionPoint originConnection,
    ConnectionPoint destinationConnection,
    bool nowCreating
)
```

Visual Basic Copy Code

```
Public Function RouteLink( _
    link As DiagramLink, _
    originConnection As ConnectionPoint, _
    destinationConnection As ConnectionPoint, _
    nowCreating As Boolean _
) As Boolean
```

Parameters

link
The DiagramLink to route.

originConnection
Represents connection to Origin node, potentially different from link's stored connection when modifying first control point.

destinationConnection
Represents connection to Destination node, potentially different from link's stored connection when modifying last control point.

nowCreating
Indicates whether link is currently being created.

Return Value

true if a path has been found successfully, otherwise false.

See Also

QuickRouter Members
QuickRouter Class
MindFusion.Diagramming Namespace

QuickRouter.RouteLinks Method

Implements ILinkRouter.RouteLinks.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection RouteLinks (
    DiagramLinkCollection links
)
```

Visual Basic Copy Code

```
Public Function RouteLinks( _
    links As DiagramLinkCollection _
) As DiagramLinkCollection
```

Parameters

links
A list of DiagramLink objects that should be routed.

Return Value

A list of links that were not routed successfully.

See Also

QuickRouter Members
QuickRouter Class
MindFusion.Diagramming Namespace

QuickRouter Properties

QuickRouter.Diagram Property

Gets or sets the diagram this router is associated with.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Diagram Diagram { get; set; }
```

Visual Basic Copy Code

```
Public Property Diagram As Diagram
```

Property Value

See Also

QuickRouter Members
QuickRouter Class
MindFusion.Diagramming Namespace

QuickRouter.Granularity Property

Gets or sets the granularity of the obstacles map generated by QuickRouter.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Granularity Granularity { get; set; }
```

Visual Basic Copy Code

```
Public Property Granularity As Granularity
```

Property Value

A member of the Granularity enumeration. The default value is **CoarseGrained**.

Remarks

A CoarseGrained obstacle map leads to faster path finding, however the found paths might be longer than necessary in a small percent of the cases. The coarse-grained map also takes slightly more time to build, so you might disable it if there are aren't many links in the diagram.

See Also

QuickRouter.TurnCost Property

Gets or sets a value added to the total cost of a route if it makes a turn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double TurnCost { get; set; }
```

Visual Basic Copy Code

```
Public Property TurnCost As Double
```

Property Value

See Also

QuickRouter Members
QuickRouter Class
MindFusion.Diagramming Namespace

QuickRouter.UBendMaxLen Property

Gets or sets the maximum U-bend length.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double UBendMaxLen { get; set; }
```

Visual Basic Copy Code

```
Public Property UBendMaxLen As Double
```

Property Value

A **double** value specifying the maximum length of U-bends. The default is 30.

See Also

QuickRouter Members
QuickRouter Class
MindFusion.Diagramming Namespace

ReadOnlyCollection(T) Class

Represents a read-only wrapper of a collection.

Syntax

C# Copy Code

```
public class ReadOnlyCollection<T> : ICollection, IEnumerable
```

Visual Basic Copy Code

```
Public Class ReadOnlyCollection(Of T)
    Implements ICollection, IEnumerable
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ReadOnlyCollection(T)
    MindFusion.Diagramming.ReadOnlyDiagramItemCollection
    MindFusion.Diagramming.ReadOnlyDiagramLinkCollection
    MindFusion.Diagramming.ReadOnlyDiagramNodeCollection

See Also

ReadOnlyCollection(T) Members
MindFusion.Diagramming Namespace

ReadOnlyCollection(T) Members

Represents a read-only wrapper of a collection.

The following tables list the members exposed by the ReadOnlyCollection(T) type.

Public Constructors

	Name	Description
	ReadOnlyCollection	Initializes a new instance of the **ReadonlyCollection** class.

Public Properties

	Name	Description
	Count	Gets the number of elements contained in the collection.
	IsSynchronized	Gets a value indicating whether the access to the collection is synchronized.
	SyncRoot	Gets an object that can be used to synchronize access to the collection.

Public Methods

	Name	Description
	CopyTo	Copies a range of elements from this collection to a compatible one-dimensional **System.Array**, starting at the specified index of the target array.
	GetEnumerator	Returns an enumerator that iterates through the collection.

See Also

ReadOnlyCollection(T) Class
MindFusion.Diagramming Namespace

ReadOnlyCollection(T) Constructor

Initializes a new instance of the ReadonlyCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ReadOnlyCollection<T> (
    ICollection<T> innerCollection
)
```

Visual Basic Copy Code

```
Public New (Of T) ( _
    innerCollection As ICollection(Of T) _
)
```

Parameters

innerCollection
A .NET **ICollection** that represents the collection to wrap.

See Also

ReadOnlyCollection(T) Class
MindFusion.Diagramming Namespace

ReadOnlyCollection(T) Methods

ReadOnlyCollection(T).CopyTo Method

Copies a range of elements from this collection to a compatible one-dimensional **System.Array**, starting at the specified index of the target array.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void CopyTo (
    Array array,
    int index
)
```

Visual Basic Copy Code

```
Public Sub CopyTo( _
    array As Array, _
    index As Integer _
)
```

Parameters

array
The one-dimensional **Array** that is the destination of the elements copied from the current collection.

index
A 32-bit integer that represents the index in array at which copying begins.

See Also

ReadOnlyCollection(T) Members
ReadOnlyCollection(T) Class
MindFusion.Diagramming Namespace

ReadOnlyCollection(T).GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IEnumerator<T> GetEnumerator ()
```

Visual Basic Copy Code

```
Public Function GetEnumerator () As IEnumerator(Of T)
```

Return Value

An .NET **IEnumerator** for the collection.

See Also

ReadOnlyCollection(T) Members
ReadOnlyCollection(T) Class
MindFusion.Diagramming Namespace

ReadOnlyCollection(T) Properties

ReadOnlyCollection(T).Count Property

Gets the number of elements contained in the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Count { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Count As Integer
```

Property Value

The number of elements actually contained in the collection.

See Also

ReadOnlyCollection(T) Members
ReadOnlyCollection(T) Class
MindFusion.Diagramming Namespace

ReadOnlyCollection(T).IsSynchronized Property

Gets a value indicating whether the access to the collection is synchronized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool IsSynchronized { get; }
```

Visual Basic

```
Public ReadOnly Property IsSynchronized As Boolean
```

Property Value

true if access to the collection is synchronized (thread safe); otherwise, **false**.

See Also

ReadOnlyCollection(T) Members
ReadOnlyCollection(T) Class
MindFusion.Diagramming Namespace

ReadOnlyCollection(T).SyncRoot Property

Gets an object that can be used to synchronize access to the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Object SyncRoot { get; }
```

Visual Basic

```
Public ReadOnly Property SyncRoot As Object
```

Property Value

An object that can be used to synchronize access to the collection.

See Also

ReadOnlyCollection(T) Members
ReadOnlyCollection(T) Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramItemCollection Class

Represents a read-only collection of items.

Syntax

C#

```
public class ReadOnlyDiagramItemCollection : ReadOnlyCollection<DiagramItem>
```

Visual Basic

```
Public Class ReadOnlyDiagramItemCollection
    Inherits ReadOnlyCollection(Of DiagramItem)
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ReadOnlyCollection(DiagramItem)
    MindFusion.Diagramming.ReadOnlyDiagramItemCollection

See Also

ReadOnlyDiagramItemCollection Members
MindFusion.Diagramming Namespace

ReadOnlyDiagramItemCollection Members

Represents a read-only collection of items.

The following tables list the members exposed by the ReadOnlyDiagramItemCollection type.

Public Constructors

	Name	Description
	ReadOnlyDiagramItemCollection	Initializes a new instance of the **ReadOnlyDiagramItemCollection** class.

Public Properties

	Name	Description
	Count	Gets the number of elements contained in the collection. (Inherited from ReadOnlyCollection.)
	IsSynchronized	Gets a value indicating whether the access to the collection is synchronized. (Inherited from ReadOnlyCollection.)
	Item	Gets the DiagramItem at the specified position.
	SyncRoot	Gets an object that can be used to synchronize access to the collection. (Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	Checks whether the specified **DiagramItem** is contained within the collection.
	CopyTo	Copies a range of elements from this collection to a compatible one-dimensional **System.Array**, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from ReadOnlyCollection.)
	IndexOf	Searches for the specified **DiagramItem** and returns the 0-based index of the first occurrence within the entire collection.

See Also

ReadOnlyDiagramItemCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramItemCollection Constructor

Initializes a new instance of the ReadOnlyDiagramItemCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ReadOnlyDiagramItemCollection (
    DiagramItemCollection innerCollection
)
```

Visual Basic Copy Code

```
Public New ( _
    innerCollection As DiagramItemCollection _
)
```

Parameters

innerCollection
A DiagramItemCollection object that represents the collection to wrap.

Remarks

A DiagramItemCollection object that represents the collection to wrap.

See Also

ReadOnlyDiagramItemCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramItemCollection Methods

ReadOnlyDiagramItemCollection.Contains Method

Checks whether the specified DiagramItem is contained within the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Contains (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Function Contains( _
    item As DiagramItem _
) As Boolean
```

Parameters

item
The DiagramItem to search for in the collection.

Return Value

true if the specified item is contained in the collection; otherwise, **false**.

See Also

ReadOnlyDiagramItemCollection Members
ReadOnlyDiagramItemCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramItemCollection.IndexOf Method

Searches for the specified DiagramItem and returns the 0-based index of the first occurrence within the entire collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int IndexOf (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Function IndexOf( _
    item As DiagramItem _
) As Integer
```

Parameters

item
The DiagramItem whose index to retrieve from the collection.

Return Value

The zero-based index of the specified item within the collection; or -1 if the item is not found.

See Also

ReadOnlyDiagramItemCollection Members
ReadOnlyDiagramItemCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramItemCollection Properties

ReadOnlyDiagramItemCollection.Item Property

Gets the DiagramItem at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem Item [
    int index
] { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Item( _
    index As Integer _
) As DiagramItem
```

Parameters

index
The zero-based index of the item to get.

Property Value

An instance of the DiagramItem class that represents the item at the specified position.

See Also

ReadOnlyDiagramItemCollection Members
ReadOnlyDiagramItemCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramLinkCollection Class

Represents a read-only collection of links.

Syntax

C# Copy Code

```
public class ReadOnlyDiagramLinkCollection : ReadOnlyCollection<DiagramLink>
```

Visual Basic Copy Code

```
Public Class ReadOnlyDiagramLinkCollection
    Inherits ReadOnlyCollection(Of DiagramLink)
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ReadOnlyCollection(DiagramLink)
    MindFusion.Diagramming.ReadOnlyDiagramLinkCollection

See Also

ReadOnlyDiagramLinkCollection Members
MindFusion.Diagramming Namespace

ReadOnlyDiagramLinkCollection Members

Represents a read-only collection of links.

The following tables list the members exposed by the ReadOnlyDiagramLinkCollection type.

Public Constructors

	Name	Description
	ReadOnlyDiagramLinkCollection	Initializes a new instance of the **ReadOnlyDiagramLinkCollection** class.

Public Properties

	Name	Description
	Count	Gets the number of elements contained in the collection. (Inherited from ReadOnlyCollection.)
	IsSynchronized	Gets a value indicating whether the access to the collection is synchronized. (Inherited from ReadOnlyCollection.)
	Item	Gets the DiagramLink at the specified position.
	SyncRoot	Gets an object that can be used to synchronize access to the collection. (Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	Checks whether the specified **DiagramLink** is contained within the collection.
	CopyTo	Copies a range of elements from this collection to a compatible one-dimensional **System.Array**, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from ReadOnlyCollection.)

	IndexOf	Searches for the specified **DiagramLink** and returns the 0-based index of the first occurrence within the entire collection.

See Also

ReadOnlyDiagramLinkCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramLinkCollection Constructor

Initializes a new instance of the ReadOnlyDiagramLinkCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ReadOnlyDiagramLinkCollection (
    DiagramLinkCollection innerCollection
)
```

Visual Basic Copy Code

```
Public New ( _
    innerCollection As DiagramLinkCollection _
)
```

Parameters

innerCollection
A DiagramLinkCollection object that represents the collection to wrap.

See Also

ReadOnlyDiagramLinkCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramLinkCollection Methods

ReadOnlyDiagramLinkCollection.Contains Method

Checks whether the specified DiagramLink is contained within the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Contains (
    DiagramLink link
)
```

Visual Basic Copy Code

```
Public Function Contains( _
    link As DiagramLink _
) As Boolean
```

Parameters

link
The DiagramLink to search for in the collection.

Return Value

true if the specified link is contained in the collection; otherwise, **false**.

See Also

ReadOnlyDiagramLinkCollection Members
ReadOnlyDiagramLinkCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramLinkCollection.IndexOf Method

Searches for the specified DiagramLink and returns the 0-based index of the first occurrence within the entire collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int IndexOf (
    DiagramLink link
)
```

Visual Basic Copy Code

```
Public Function IndexOf( _
    link As DiagramLink _
) As Integer
```

Parameters

link
The DiagramLink whose index to retrieve from the collection.

Return Value

The zero-based index of the specified link within the collection; or -1 if the link is not found.

See Also

ReadOnlyDiagramLinkCollection Members
ReadOnlyDiagramLinkCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramLinkCollection Properties

ReadOnlyDiagramLinkCollection.Item Property

Gets the DiagramLink at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink Item [
    int index
] { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Item( _
    index As Integer _
) As DiagramLink
```

Parameters

index
The zero-based index of the link to get.

Property Value

An instance of the DiagramLink class that represents the link at the specified position.

See Also

ReadOnlyDiagramLinkCollection Members
ReadOnlyDiagramLinkCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramNodeCollection Class

Represents a read-only collection of nodes.

Syntax

C# Copy Code

```
public class ReadOnlyDiagramNodeCollection : ReadOnlyCollection<DiagramNode>
```

Visual Basic Copy Code

```
Public Class ReadOnlyDiagramNodeCollection
    Inherits ReadOnlyCollection(Of DiagramNode)
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ReadOnlyCollection(DiagramNode)
    MindFusion.Diagramming.ReadOnlyDiagramNodeCollection

See Also

ReadOnlyDiagramNodeCollection Members
MindFusion.Diagramming Namespace

ReadOnlyDiagramNodeCollection Members

Represents a read-only collection of nodes.

The following tables list the members exposed by the ReadOnlyDiagramNodeCollection type.

Public Constructors

	Name	Description
	ReadOnlyDiagramNodeCollection	Initializes a new instance of the **ReadOnlyDiagramNodeCollection** class.

Public Properties

	Name	Description
	Count	Gets the number of elements contained in the collection. (Inherited from ReadOnlyCollection.)
	IsSynchronized	Gets a value indicating whether the access to the collection is synchronized. (Inherited from ReadOnlyCollection.)

	Item	Gets the DiagramNode at the specified position.
	SyncRoot	Gets an object that can be used to synchronize access to the collection. (Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	Checks whether the specified **DiagramNode** is contained within the collection.
	CopyTo	Copies a range of elements from this collection to a compatible one-dimensional **System.Array**, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from ReadOnlyCollection.)
	IndexOf	Searches for the specified **DiagramNode** and returns the 0-based index of the first occurrence within the entire collection.

See Also

ReadOnlyDiagramNodeCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramNodeCollection Constructor

Initializes a new instance of the ReadOnlyDiagramNodeCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ReadOnlyDiagramNodeCollection (
    DiagramNodeCollection innerCollection
)
```

Visual Basic Copy Code

```
Public New ( _
    innerCollection As DiagramNodeCollection _
)
```

Parameters

innerCollection
A DiagramNodeCollection object that represents the collection to wrap.

See Also

ReadOnlyDiagramNodeCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramNodeCollection Methods

ReadOnlyDiagramNodeCollection.Contains Method

Checks whether the specified DiagramNode is contained within the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Contains (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Function Contains( _
    node As DiagramNode _
) As Boolean
```

Parameters

node
The DiagramNode to search for in the collection.

Return Value

true if the specified node is contained in the collection; otherwise, **false**.

See Also

ReadOnlyDiagramNodeCollection Members
ReadOnlyDiagramNodeCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramNodeCollection.IndexOf Method

Searches for the specified DiagramNode and returns the 0-based index of the first occurrence within the entire collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int IndexOf (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Function IndexOf( _
    node As DiagramNode _
) As Integer
```

Parameters

node
The DiagramNode whose index to retrieve from the collection.

Return Value

The zero-based index of the specified node within the collection; or -1 if the node is not found.

See Also

ReadOnlyDiagramNodeCollection Members
ReadOnlyDiagramNodeCollection Class
MindFusion.Diagramming Namespace

ReadOnlyDiagramNodeCollection Properties

ReadOnlyDiagramNodeCollection.Item Property

Gets the DiagramNode at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNode Item [
    int index
] { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Item( _
    index As Integer _
) As DiagramNode
```

Parameters

index
The zero-based index of the node to get.

Property Value

An instance of the DiagramNode class that represents the node at the specified position.

See Also

ReadOnlyDiagramNodeCollection Members
ReadOnlyDiagramNodeCollection Class
MindFusion.Diagramming Namespace

Relationship Enumeration

Specifies the type of relationship between two tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum Relationship
```

Visual Basic Copy Code

```
Public Enum Relationship
```

Members

	Member name	Description
	OneToMany	The relationship is of type 1:M, that is, one to many.
	ManyToOne	The relationship is of type M:1, that is, many to one.

Remarks

Use the enumeration values when drawing database diagrams. It helps you specify how database tables' fields are related.

See Also

MindFusion.Diagramming Namespace
AddRelation Method

RelativeToLink Enumeration

Specifies how the link labels are positioned.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum RelativeToLink
```

Visual Basic Copy Code

```
Public Enum RelativeToLink
```

Members

	Member name	Description
	Segment	Indicates that the link labels are positioned relative to a specified segment.
	ControlPoint	Indicates that the link labels are positioned relative to a specified control point.
	LinkLength	Indicates that the link labels are positioned relative to the link.

See Also

MindFusion.Diagramming Namespace
LinkLabel.RelativeTo Property

RelativeToNode Enumeration

Specifies how the node labels are positioned.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum RelativeToNode
```

Visual Basic Copy Code

```
Public Enum RelativeToNode
```

Members

	Member name	Description
	Edge	Indicates that the node labels are positioned relative to a specified edge.
	Corner	Indicates that the node labels are positioned relative to a specified corner point.

	Center	Indicates that the node labels are positioned relative to the node center.

See Also

MindFusion.Diagramming Namespace

RenderOptions Class

Provides properties that allow customization of various aspects of printing and print-preview.

Syntax

C# Copy Code

```
public class RenderOptions
```

Visual Basic Copy Code

```
Public Class RenderOptions
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.RenderOptions

See Also

RenderOptions Members
MindFusion.Diagramming Namespace

RenderOptions Members

Provides properties that allow customization of various aspects of printing and print-preview.

The following tables list the members exposed by the RenderOptions type.

Public Constructors

	Name	Description
	RenderOptions	Overloaded.

Public Properties

	Name	Description
	EnableAnchors	Gets or sets a value indicating whether the printing of anchor point marks is enabled.
	EnableBackground	Gets or sets a value indicating whether the printing of the background color is enabled.
	EnableBackgroundImage	Gets or sets a value indicating whether the printing of the background image is enabled.
	EnableEffects	Enables or disables special effects, such as emboss.
	EnableGrid	Gets or sets a value indicating whether the alignment grid is rendered.
	EnableImages	Gets or sets a value indicating whether the printing of images that are displayed in nodes is enabled.
	EnableInterior	Gets or sets a value indicating whether the printing of node interiors is enabled.

	EnableShadows	Gets or sets a value indicating whether the printing of the shadows of diagram items is enabled.
	EnableText	Gets or sets a value indicating whether the printing of the text of diagram items is enabled.
	HyperLinkStyle	Specifies how various export components should treat the HyperLink property of items.
	HyperLinkTarget	Specify the value of the **target** attribute generated by various export components for the **HyperLink** property of items.
	InternalMsg	For internal use.
	MeasureUnit	Specifies the diagram's unit of measure.
	PaintControls	Gets or sets a value indicating whether to execute custom painting code for **ControlNode** items when printing.
	RenderTarget	Gets or sets the rendering target.
	Scale	Gets or sets the scale of the print output relatively to the original on-screen size of the diagram.
	ScrollX	Gets or sets the document's X scroll position.
	ScrollY	Gets or sets the document's Y scroll position.
	SmoothingMode	Gets or sets the current anti-aliasing mode.
	TargetView	Gets or sets the object that currently displays the Diagram.
	TextRenderingHint	Gets or sets the rendering mode for text displayed in this diagram.
	VisibleRect	Specifies which part of the diagram is currently visible.

See Also

RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions Constructor

Overload List

Name	Description
RenderOptions ()	Initializes a new instance of the RenderOptions class.
RenderOptions (RenderOptions)	Initializes a new instance of the **RenderOptions** class using the specified instance as a prototype.

See Also

RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions Constructor ()

Initializes a new instance of the RenderOptions class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RenderOptions ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

RenderOptions Constructor Overload List
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions Constructor (RenderOptions)

Initializes a new instance of the RenderOptions class using the specified instance as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RenderOptions (
    RenderOptions prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As RenderOptions _
)
```

Parameters

prototype
The RenderOptions instance to use as a prototype.

See Also

RenderOptions Constructor Overload List
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions Properties

RenderOptions.EnableAnchors Property

Gets or sets a value indicating whether the printing of anchor point marks is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableAnchors { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableAnchors As Boolean
```

Property Value

true if anchor point marks should be printed; otherwise, **false**. The default is **false**.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.EnableBackground Property

Gets or sets a value indicating whether the printing of the background color is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableBackground { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableBackground As Boolean
```

Property Value

true if the diagram background should be printed; otherwise, **false**. The default is **false**.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.EnableBackgroundImage Property

Gets or sets a value indicating whether the printing of the background image is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableBackgroundImage { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableBackgroundImage As Boolean
```

Property Value

true if the background image should be printed; otherwise, **false**. The default is **false**.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.EnableEffects Property

Enables or disables special effects, such as emboss.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableEffects { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableEffects As Boolean
```

Property Value

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.EnableGrid Property

Gets or sets a value indicating whether the alignment grid is rendered.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableGrid { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableGrid As Boolean
```

Property Value

true to render the alignment grid, if visible; otherwise, **false**.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.EnableImages Property

Gets or sets a value indicating whether the printing of images that are displayed in nodes is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableImages { get; set; }
```

Visual Basic

```
Public Property EnableImages As Boolean
```

Property Value

true to print node images; otherwise, **false**.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.EnableInterior Property

Gets or sets a value indicating whether the printing of node interiors is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool EnableInterior { get; set; }
```

Visual Basic

```
Public Property EnableInterior As Boolean
```

Property Value

true to print node interiors; otherwise, **false**.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.EnableShadows Property

Gets or sets a value indicating whether the printing of the shadows of diagram items is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool EnableShadows { get; set; }
```

Visual Basic

```
Public Property EnableShadows As Boolean
```

Property Value

true to print item shadows; otherwise, **false**.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.EnableText Property

Gets or sets a value indicating whether the printing of the text of diagram items is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableText { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableText As Boolean
```

Property Value

true to print item texts; otherwise, **false**.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.HyperLinkStyle Property

Specifies how various export components should treat the HyperLink property of items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HyperLinkStyle HyperLinkStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property HyperLinkStyle As HyperLinkStyle
```

Property Value

A member of the HyperLinkStyle enumeration.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.HyperLinkTarget Property

Specify the value of the **target** attribute generated by various export components for the HyperLink property of items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string HyperLinkTarget { get; set; }
```

Visual Basic

```
Public Property HyperLinkTarget As String
```

Property Value

The default value is "_self".

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.InternalMsg Property

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static string InternalMsg { get; set; }
```

Visual Basic

```
Public Shared Property InternalMsg As String
```

Property Value

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.MeasureUnit Property

Specifies the diagram's unit of measure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public MeasureUnit MeasureUnit { get; set; }
```

Visual Basic

```
Public Property MeasureUnit As MeasureUnit
```

Property Value

An instance of the MeasureUnit class specifying the diagram's unit of measure.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.PaintControls Property

Gets or sets a value indicating whether to execute custom painting code for **ControlNode** items when printing.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool PaintControls { get; set; }
```

Visual Basic Copy Code

```
Public Property PaintControls As Boolean
```

Property Value

true to raise the appropriate event when printing controls; otherwise, **false**.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.RenderTarget Property

Gets or sets the rendering target.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RenderTarget RenderTarget { get; set; }
```

Visual Basic Copy Code

```
Public Property RenderTarget As RenderTarget
```

Property Value

A value from the RenderTarget enumeration specifying whether rendering is done on the screen or on a printer.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.Scale Property

Gets or sets the scale of the print output relatively to the original on-screen size of the diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Scale { get; set; }
```

Visual Basic

```
Public Property Scale As Double
```

Property Value

A **double** value specifying the scale of the print output as percentage of original size.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.ScrollX Property

Gets or sets the document's X scroll position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double ScrollX { get; set; }
```

Visual Basic

```
Public Property ScrollX As Double
```

Property Value

A **double** value specifying the X scroll position.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.ScrollY Property

Gets or sets the document's Y scroll position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double ScrollY { get; set; }
```

Visual Basic

```
Public Property ScrollY As Double
```

Property Value

A **double** value specifying the Y scroll position.

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.SmoothingMode Property

Gets or sets the current anti-aliasing mode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SmoothingMode SmoothingMode { get; set; }
```

Visual Basic Copy Code

```
Public Property SmoothingMode As SmoothingMode
```

Property Value

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.TargetView Property

Gets or sets the object that currently displays the Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object TargetView { get; set; }
```

Visual Basic Copy Code

```
Public Property TargetView As Object
```

Property Value

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.TextRenderingHint Property

Gets or sets the rendering mode for text displayed in this diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TextRenderingHint TextRenderingHint { get; set; }
```

Visual Basic Copy Code

```
Public Property TextRenderingHint As TextRenderingHint
```

Property Value

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderOptions.VisibleRect Property

Specifies which part of the diagram is currently visible.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect VisibleRect { get; set; }
```

Visual Basic Copy Code

```
Public Property VisibleRect As Rect
```

Property Value

See Also

RenderOptions Members
RenderOptions Class
MindFusion.Diagramming Namespace

RenderTarget Enumeration

Specifies where the diagram is being rendered.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum RenderTarget
```

Visual Basic Copy Code

```
Public Enum RenderTarget
```

Members

	Member name	Description
	Display	The diagram is rendered on the screen.
	Printer	The diagram is rendered on a printer.
	BackBuffer	The diagram is rendered on offscreen buffer bitmap.

See Also

MindFusion.Diagramming Namespace

RepaintEventArgs Class

Specifies the coordinates of the area that should be repainted by views when the Repaint event is raised.

Syntax

C# Copy Code

```
public class RepaintEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class RepaintEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.RepaintEventArgs

See Also

RepaintEventArgs Members
MindFusion.Diagramming Namespace

RepaintEventArgs Members

Specifies the coordinates of the area that should be repainted by views when the Repaint event is raised.

The following tables list the members exposed by the RepaintEventArgs type.

Public Constructors

	Name	Description
	RepaintEventArgs	Overloaded. Initializes a new instance of the **RepaintEventArgs** class.

Public Properties

	Name	Description
	InvalidRect	Gets the rectangular area in the diagram document that should be repainted.
	RedrawCacheImage	Gets a value indicating whether the back-buffer cache image should be repainted as well.
	SourceView	Gets the view object that raised the event.

See Also

RepaintEventArgs Class
MindFusion.Diagramming Namespace

RepaintEventArgs Constructor

Initializes a new instance of the RepaintEventArgs class.

Overload List

Name	Description
RepaintEventArgs (Rect)	Initializes a new instance of the **RepaintEventArgs** class.
RepaintEventArgs (Rect, Boolean)	Initializes a new instance of the **RepaintEventArgs** class.
RepaintEventArgs (Rect, Object)	Initializes a new instance of the **RepaintEventArgs** class.

See Also

RepaintEventArgs Class
MindFusion.Diagramming Namespace

RepaintEventArgs Constructor (Rect)

Initializes a new instance of the RepaintEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RepaintEventArgs (
    Rect invalidRect
)
```

Visual Basic Copy Code

```
Public New ( _
    invalidRect As Rect _
)
```

Parameters

invalidRect
A **Microsoft.Maui.Graphics.Rect** object representing the rectangle that should be repainted.

See Also

RepaintEventArgs Constructor Overload List
RepaintEventArgs Class
MindFusion.Diagramming Namespace

RepaintEventArgs Constructor (Rect, Boolean)

Initializes a new instance of the RepaintEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RepaintEventArgs (
    Rect invalidRect,
    bool redrawCacheImage
)
```

Visual Basic Copy Code

```
Public New ( _
    invalidRect As Rect, _
    redrawCacheImage As Boolean _
)
```

Parameters

invalidRect

A **Microsoft.Maui.Graphics.Rect** object representing the rectangle that should be repainted.

redrawCacheImage

A boolean value specifying whether the diagram cache image used during modification should be re-created.

See Also

RepaintEventArgs Constructor Overload List
RepaintEventArgs Class
MindFusion.Diagramming Namespace

RepaintEventArgs Constructor (Rect, Object)

Initializes a new instance of the RepaintEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RepaintEventArgs (
    Rect invalidRect,
    Object sourceView
)
```

Visual Basic Copy Code

```
Public New ( _
    invalidRect As Rect, _
    sourceView As Object _
)
```

Parameters

invalidRect

A **Microsoft.Maui.Graphics.Rect** object representing the rectangle that should be repainted.

sourceView

The view object that raised the event.

See Also

RepaintEventArgs Constructor Overload List
RepaintEventArgs Class
MindFusion.Diagramming Namespace

RepaintEventArgs Properties

RepaintEventArgs.InvalidRect Property

Gets the rectangular area in the diagram document that should be repainted.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect InvalidRect { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property InvalidRect As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** instance specifying the invalid rectangle. The value of **Empty** indicates that the whole view should be repainted.

See Also

RepaintEventArgs Members
RepaintEventArgs Class
MindFusion.Diagramming Namespace

RepaintEventArgs.RedrawCacheImage Property

Gets a value indicating whether the back-buffer cache image should be repainted as well.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RedrawCacheImage { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property RedrawCacheImage As Boolean
```

Property Value

true to redraw the back-buffer image; otherwise, **false**.

See Also

RepaintEventArgs Members
RepaintEventArgs Class
MindFusion.Diagramming Namespace

RepaintEventArgs.SourceView Property

Gets the view object that raised the event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object SourceView { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property SourceView As Object
```

Property Value

An object identifying the view that raised the event.

See Also

RepaintEventArgs Members
RepaintEventArgs Class
MindFusion.Diagramming Namespace

RerouteLinks Enumeration

Provides values indicating in what circumstances links must be rerouted automatically.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum RerouteLinks
```

Visual Basic Copy Code

```
Public Enum RerouteLinks
```

Members

	Member name	Description
	Never	Do not reroute links automatically.
	WhenModified	Reroute a link if its end points are moved around.
	WhenIntersectNode	Reroute a link if it intersects a node.
	WhileCreating	Route new links while users draw them. This flag can be combined with any of the previous two flags.
	WhileModifying	Route links while the user modifies them or drags their incident nodes.

Remarks

These values can be assigned to the TriggerRerouting property of RoutingOptions.

See Also

MindFusion.Diagramming Namespace

RestrictToBounds Enumeration

Specifies a restriction of the position of items relatively to the diagram bounds.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum RestrictToBounds
```

Visual Basic Copy Code

```
Public Enum RestrictToBounds
```

Members

	Member name	Description
	NoRestriction	No restriction on the position of items.
	Intersection	Items must be located at least partially inside the document extents.
	InsideOnly	Items must be located entirely inside the document extents.

Remarks

This enumeration specifies the possible values of the RestrictItemsToBounds property.

See Also

MindFusion.Diagramming Namespace
RestrictItemsToBounds Property

RoundRectangleTemplate Class

Represents an ElementTemplate that draws round rectangle. This class ensures that the round corners do not change during resizing which is visually better. The drawback is that the radius appears to be specified in DiagramView Measure units and therefore a given shape instance will look bad when the measure unit is not the one for which the shape was designed.

Syntax

C# Copy Code

```
public class RoundRectangleTemplate : ElementTemplate
```

Visual Basic Copy Code

```
Public Class RoundRectangleTemplate
    Inherits ElementTemplate
```

Remarks

This class ensures that the round corners do not change during resizing which is visually better. The drawback is that the radius appears to be specified in DiagramView Measure units and therefore a given shape instance will look bad when the measure unit is not the one for which the shape was designed.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ElementTemplate
    MindFusion.Diagramming.RoundRectangleTemplate

See Also

RoundRectangleTemplate Members
MindFusion.Diagramming Namespace

RoundRectangleTemplate Members

Represents an ElementTemplate that draws round rectangle. This class ensures that the round corners do not change during resizing which is visually better. The drawback is that the radius appears to be specified in DiagramView Measure units and therefore a given shape instance will look bad when the measure unit is not the one for which the shape was designed.

The following tables list the members exposed by the RoundRectangleTemplate type.

Public Constructors

	Name	Description
	RoundRectangleTemplate	Overloaded. Initializes a new instance of the **RoundRectangleTemplate** class.

Public Properties

	Name	Description
	Bounds	Gets the bounding rectangle of this element.
	Color	Gets or sets the outline color of this element. (Inherited from ElementTemplate.)
	DashStyle	Gets or sets the dash style of the outline of this element. (Inherited from ElementTemplate.)
	Radius	Gets or sets the round radius of this element.
	Width	Gets or sets the outline width of this element. (Inherited from ElementTemplate.)

Public Methods

	Name	Description
	Clone	Creates an exact copy of this **RoundRectangleTemplate** element.
	SameAs	ElementTemplate.SameAs override.

Protected Methods

	Name	Description
	GetElementPen	Gets a pen used to render the element. If no pen properties are explicitly specified for this element, returns the pen passed as an argument. (Inherited from ElementTemplate.)

See Also

RoundRectangleTemplate Class
MindFusion.Diagramming Namespace

RoundRectangleTemplate Constructor

Initializes a new instance of the RoundRectangleTemplate class.

Overload List

Name	Description
RoundRectangleTemplate ()	Initializes a new instance of the **RoundRectangleTemplate** class.
RoundRectangleTemplate (Double, Double, Double, Double, Double)	Initializes a new instance of the **RoundRectangleTemplate** class with the specified settings.
RoundRectangleTemplate (Double, Double, Double, Double, Double, Color, DashStyle, Double)	Initializes a new instance of the RoundRectangleTemplate class with the specified settings.

See Also

RoundRectangleTemplate Class
MindFusion.Diagramming Namespace

RoundRectangleTemplate Constructor ()

Initializes a new instance of the RoundRectangleTemplate class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RoundRectangleTemplate ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

RoundRectangleTemplate Constructor Overload List
RoundRectangleTemplate Class
MindFusion.Diagramming Namespace

RoundRectangleTemplate Constructor (Double, Double, Double, Double, Double)

Initializes a new instance of the RoundRectangleTemplate class with the specified settings.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RoundRectangleTemplate (
    double x,
    double y,
    double w,
    double h,
    double r
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Double, _
    y As Double, _
    w As Double, _
    h As Double, _
    r As Double _
)
```

Parameters

x
The x-coordinate of the top-left corner of the rectangle, expressed as percentage of a shape node's bounding rectangle.

y
The y-coordinate of the top-left corner of the rectangle, expressed as percentage of a shape node's bounding rectangle.

w
The width of the rectangle, expressed as percentage of a shape node's bounding rectangle.

h
The height of the rectangle, expressed as percentage of a shape node's bounding rectangle.

r
The round radius of the rectangle.

See Also

RoundRectangleTemplate Constructor (Double, Double, Double, Double, Double, Color, DashStyle, Double)

Initializes a new instance of the RoundRectangleTemplate class with the specified settings.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RoundRectangleTemplate (
    double x,
    double y,
    double w,
    double h,
    double r,
    Color color,
    DashStyle dashStyle,
    double width
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Double, _
    y As Double, _
    w As Double, _
    h As Double, _
    r As Double, _
    color As Color, _
    dashStyle As DashStyle, _
    width As Double _
)
```

Parameters

x
The x-coordinate of the top-left corner of the rectangle, expressed as percentage of a shape node's bounding rectangle.

y
The y-coordinate of the top-left corner of the rectangle, expressed as percentage of a shape node's bounding rectangle.

w
The width of the rectangle, expressed as percentage of a shape node's bounding rectangle.

h
The height of the rectangle, expressed as percentage of a shape node's bounding rectangle.

r
The round radius of the rectangle.

color
The color of this segment.

dashStyle
The dash style of this segment.

width
The width of this segment.

See Also

RoundRectangleTemplate Methods

RoundRectangleTemplate.Clone Method

Creates an exact copy of this RoundRectangleTemplate element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Object Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Object
```

Return Value

An instance of the RoundRectangleTemplate class.

See Also

RoundRectangleTemplate Members
RoundRectangleTemplate Class
MindFusion.Diagramming Namespace

RoundRectangleTemplate.SameAs Method

ElementTemplate.SameAs override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool SameAs (
    ElementTemplate obj
)
```

Visual Basic Copy Code

```
Public Overrides Function SameAs( _
    obj As ElementTemplate _
) As Boolean
```

Parameters

obj
The ElementTemplate to compare to this template.

Return Value

true if the specified object is a RoundRectangleTemplate closely approximating this template; otherwise, **false**.

See Also

RoundRectangleTemplate Members
RoundRectangleTemplate Class
MindFusion.Diagramming Namespace

RoundRectangleTemplate Properties

RoundRectangleTemplate.Bounds Property

Gets the bounding rectangle of this element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect Bounds { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Bounds As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** that represents the bounding rectangle of this element.

See Also

RoundRectangleTemplate Members
RoundRectangleTemplate Class
MindFusion.Diagramming Namespace

RoundRectangleTemplate.Radius Property

Gets or sets the round radius of this element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Radius { get; set; }
```

Visual Basic Copy Code

```
Public Property Radius As Double
```

Property Value

A **double** value specifying the round radius of this element.

See Also

RoundRectangleTemplate Members
RoundRectangleTemplate Class
MindFusion.Diagramming Namespace

RoutePattern Class

Represents a template from which routes are created.

Syntax

C# Copy Code

```
public class RoutePattern
```

Visual Basic Copy Code

```
Public Class RoutePattern
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.RoutePattern

See Also

RoutePattern Members
MindFusion.Diagramming Namespace

RoutePattern Members

Represents a template from which routes are created.

The following tables list the members exposed by the RoutePattern type.

Public Constructors

	Name	Description
	RoutePattern	Overloaded.

Public Properties

	Name	Description
	Steps	A list of RouteStep objects that generate a route.

See Also

RoutePattern Class
MindFusion.Diagramming Namespace

RoutePattern Constructor

Overload List

Name	Description
RoutePattern ()	Initializes a new instance of the RoutePattern class.
RoutePattern (String)	Initializes a new instance of the RoutePattern class.

See Also

RoutePattern Class
MindFusion.Diagramming Namespace

RoutePattern Constructor ()

Initializes a new instance of the RoutePattern class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RoutePattern ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

RoutePattern Constructor Overload List
RoutePattern Class
MindFusion.Diagramming Namespace

RoutePattern Constructor (String)

Initializes a new instance of the RoutePattern class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RoutePattern (
    string data
)
```

Visual Basic Copy Code

```
Public New ( _
    data As String _
)
```

Parameters

data
A string representing a sequence of RouteSteps.

Remarks

Members of the RouteStepKind enumeration are represented by three-character acronyms, with a numerical value appended to advance- steps.

Example

The following route can be represented by "ADR0.5 TRR ADR1 TRL ADR0.5" string.

C# Copy Code

```
var rightZigzag = new RoutePattern();
rightZigzag.Steps.Add(
    new RouteStep(RouteStepKind.AdvanceRelative, 0.5f));
rightZigzag.Steps.Add(
    new RouteStep(RouteStepKind.TurnRight));
rightZigzag.Steps.Add(
    new RouteStep(RouteStepKind.AdvanceRelative, 1));
rightZigzag.Steps.Add(
    new RouteStep(RouteStepKind.TurnLeft));
rightZigzag.Steps.Add(
    new RouteStep(RouteStepKind.AdvanceRelative, 0.5f));
router.Patterns.Add(rightZigzag);

// this can be defined in shorter form using a pattern string
// var rightZigzag = new RoutePattern(
//     "ADR0.5 TRR ADR1 TRL ADR0.5");
```

See Also

RoutePattern Constructor Overload List
RoutePattern Class
MindFusion.Diagramming Namespace

RoutePattern Properties

RoutePattern.Steps Property

A list of RouteStep objects that generate a route.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<RouteStep> Steps { get; set; }
```

Visual Basic Copy Code

```
Public Property Steps As List(Of RouteStep)
```

Property Value

See Also

RoutePattern Members
RoutePattern Class
MindFusion.Diagramming Namespace

RouteStep Class

Represents segments or turns in routing patterns.

Syntax

C# Copy Code

```
public class RouteStep
```

Visual Basic Copy Code

```
Public Class RouteStep
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.RouteStep

See Also

RouteStep Members
MindFusion.Diagramming Namespace

RouteStep Members

Represents segments or turns in routing patterns.

The following tables list the members exposed by the RouteStep type.

Public Constructors

	Name	Description
	RouteStep	Overloaded.

See Also

RouteStep Class
MindFusion.Diagramming Namespace

RouteStep Constructor

Overload List

Name	Description
RouteStep (RouteStepKind)	Initializes a new instance of the RouteStep class.
RouteStep (RouteStepKind, Double)	Initializes a new instance of the **RouteStep** class.

See Also

RouteStep Class
MindFusion.Diagramming Namespace

RouteStep Constructor (RouteStepKind)

Initializes a new instance of the RouteStep class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RouteStep (
    RouteStepKind kind
)
```

Visual Basic Copy Code

```
Public New ( _
    kind As RouteStepKind _
)
```

Parameters

kind
A member of the RouteStepKind enumeration.

See Also

RouteStep Constructor Overload List
RouteStep Class
MindFusion.Diagramming Namespace

RouteStep Constructor (RouteStepKind, Double)

Initializes a new instance of the RouteStep class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RouteStep (
    RouteStepKind kind,
    float amount
)
```

Visual Basic Copy Code

```
Public New ( _
    kind As RouteStepKind, _
    amount As Single _
)
```

Parameters

kind
A member of the RouteStepKind enumeration.

amount
The distance to add when step kind is Advance*.

See Also

RouteStep Constructor Overload List
RouteStep Class
MindFusion.Diagramming Namespace

RouteStepKind Enumeration

Represents parts of a route template.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum RouteStepKind
```

Visual Basic Copy Code

```
Public Enum RouteStepKind
```

Members

	Member name	Description
	AdvanceRelative	Advance by the specified relative distance from origin to destination node. Equivalent to "ADR" command in pattern strings.
	AdvanceAbsolute	Advance by the specified absolute distance. Corresponds to "ADA" command in pattern strings.
	AdvancePreferred	Advance by PreferredDistance.
	AdvanceDest	Advance by the specified relative remaining distance to destination node. Corresponds to "ADD" command in pattern strings.

	AdvanceRemaining	Advance by the specified relative remaining distance to link's end point. Corresponds to "ADM" command in pattern strings.
	AlignToOrigin	Advance to align with nearest side of origin node. Corresponds to "ALO" command in pattern strings.
	AlignToDestination	Advance to align with nearest side of destination node. Corresponds to "ALD" command in pattern strings.
	AlignToStart	Advance to align with link's start point. Corresponds to "ALS" command in pattern strings.
	AlignToEnd	Advance to align with link's end point. Corresponds to "ALE" command in pattern strings.
	AlignToFarBorder	Advance to align with furthest border of both nodes. Corresponds to "AFB" command in pattern strings.
	TurnLeft	Turn 90 degrees to the left. Corresponds to "TRL" command in pattern strings.
	TurnRight	Turn 90 degrees to the right. Corresponds to "TRR" command in pattern strings.

See Also

MindFusion.Diagramming Namespace

RoutingOptions Class

Allows customization of various aspects of the link routing algorithm.

Syntax

C# Copy Code

```
public class RoutingOptions : RoutingOptions
```

Visual Basic Copy Code

```
Public Class RoutingOptions
    Implements RoutingOptions
```

Remarks

Use the RoutingOptions member of the Diagram class to access the sole instance of this class.

MvcDiagram uses a cost minimization algorithm to route links between nodes. For each link, there is a routing grid built for the area that encloses the link and its origin and destination nodes. The size of the grid pieces is set via GridSize. Each piece of the grid is marked as an obstacle if it intersects an **Obstacle** node. Some cost values are assigned to the remaining pieces, depending on how distant they are from nodes and other links.

That algorithm evaluates possible paths across the routing grid aiming to find one that has a minimal cost. The values of LengthCost, TurnCost, CrossingCost and NodeVicinityCost are added to the total cost of a route and have a great influence over the shape of the chosen path.

The StartOrientation and EndOrientation properties specify the orientation of the first and last segments of routed links, and Anchoring specifies whether link ends should be aligned to the anchor points defined for nodes.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.RoutingOptions

See Also

RoutingOptions Members
MindFusion.Diagramming Namespace

RoutingOptions Members

Allows customization of various aspects of the link routing algorithm.

The following tables list the members exposed by the RoutingOptions type.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating whether to align link ends to the anchor points of nodes.
	AvoidOverlaps	Gets or sets whether link segments should be pulled apart when they overlap in initially found routes.
	CrossingCost	Gets or sets a value added to the total cost of a route which causes a link to cross another link.
S	Distance	Gets a heuristic delegate function that returns the real distance sqrt(dx^2 + dy^2) between two pieces of the routing grid.
S	DistSquare	Gets a heuristic delegate function that returns the square distance dx^2 + dy^2 as a cost evaluation of the distance between two pieces of the routing grid.
	DontOptimizeLongRoutes	Gets or sets a value indicating whether long routes don't have their number of segments minimized in order to save up processing time.
	EndOrientation	Gets or sets the desired orientation of the last segments of routed links.
	EndPointsOffset	Gets or sets the minimum number of routing grid cells the end segments of links should occupy before they are allowed to bend.
	GridSize	Gets or sets the size of the pieces of the routing grid.
	IgnoreAnchoring	For internal use.
	LengthCost	Gets or sets a value added to the total cost of a route for each piece of the routing grid occupied by the route.
S	MaxAbs	Gets a heuristic delegate function that returns max(dx, dy) as a cost evaluation of the distance between two pieces of the routing grid.
	NodeVicinityCost	Gets or sets a value added to the total cost of a route if the route passes too close to a node.
	NodeVicinitySize	Gets or sets the size of a buffer zone around a node for which routes get a penalty cost if they pass through it.
	PolylineRouting	Gets or sets a value indicating whether the segments of Polyline links should be oriented orthogonally to node's borders and each other.
	RouteHeuristics	Gets or sets a heuristic function used to evaluate the distance between two pieces of the routing grid.
	StartOrientation	Gets or sets the desired orientation of the first segments of routed links.
	TriggerRerouting	Gets or sets a value indicating under what circumstances links are rerouted automatically.
	TurnCost	Gets or sets a value added to the total cost of a route if it makes a turn.

Public Methods

	Name	Description
	SetDefaultMode	Resets all routing properties to their default values.
	SetMinSegmentsMode	Assigns to the routing properties a predefined set of values which allow finding paths that have a small number of segments.

	SetQuickMode	Assigns to the routing properties a predefined set of values which allow finding paths quickly but the found paths are of low quality.

See Also

RoutingOptions Class
MindFusion.Diagramming Namespace

RoutingOptions Methods

RoutingOptions.SetDefaultMode Method

Resets all routing properties to their default values.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetDefaultMode ()
```

Visual Basic Copy Code

```
Public Sub SetDefaultMode ()
```

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
SetMinSegmentsMode Method
SetQuickMode Method

RoutingOptions.SetMinSegmentsMode Method

Assigns to the routing properties a predefined set of values which allow finding paths that have a small number of segments.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetMinSegmentsMode ()
```

Visual Basic Copy Code

```
Public Sub SetMinSegmentsMode ()
```

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
SetDefaultMode Method
SetQuickMode Method

RoutingOptions.SetQuickMode Method

Assigns to the routing properties a predefined set of values which allow finding paths quickly but the found paths are of low quality.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SetQuickMode ()
```

Visual Basic Copy Code

```
Public Sub SetQuickMode ()
```

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
SetDefaultMode Method
SetMinSegmentsMode Method

RoutingOptions Properties

RoutingOptions.Anchoring Property

Gets or sets a value indicating whether to align link ends to the anchor points of nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Anchoring Anchoring { get; set; }
```

Visual Basic Copy Code

```
Public Property Anchoring As Anchoring
```

Property Value

A member of the Anchoring enumeration. The default is **Ignore**.

Remarks

By default, the routing algorithm ignores anchor points. To make it align link ends to the points specified in the **AnchorPattern** property of nodes, set **Anchoring** to Keep or **Reassign**. **Keep** specifies that the routing algorithm does not change the positions of link end points. **Reassign** lets the routing algorithm choose new end points for a link from the ones specified in the **AnchorPattern** of its origin and destination nodes.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace

RoutingOptions.AvoidOverlaps Property

Gets or sets whether link segments should be pulled apart when they overlap in initially found routes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AvoidOverlaps { get; set; }
```

Visual Basic Copy Code

```
Public Property AvoidOverlaps As Boolean
```

Property Value

true to pull apart overlapping segments, or **false** otherwise. The default value is **true**.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace

RoutingOptions.CrossingCost Property

Gets or sets a value added to the total cost of a route which causes a link to cross another link.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public byte CrossingCost { get; set; }
```

Visual Basic Copy Code

```
Public Property CrossingCost As Byte
```

Property Value

A byte value specifying the crossing cost. The default is 4.

Remarks

This is a penalty value added to the total cost of an evaluated link route if it crosses another link. The routing algorithm aims to find paths that have a minimal cost, so assigning a greater value to **CrossingCost** leads to fewer link intersections. At this time **CrossingCost** processing is implemented only for links with Shape set to Cascading.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
LengthCost Property
NodeVicinityCost Property
TurnCost Property

RoutingOptions.Distance Property

Gets a heuristic delegate function that returns the real distance sqrt(dx^2 + dy^2) between two pieces of the routing grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static RouteHeuristics Distance { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Distance As RouteHeuristics
```

Property Value

A predefined RouteHeuristics delegate instance.

Remarks

While generating a path, the routing algorithm evaluates pieces of the routing grid adjacent to the current one. It takes into account the cost for passing trough a piece and adds to that cost a heuristic value calculated by a RouteHeuristics delegate. So the smaller heuristic value a piece has, the greater the chance the found route to go through that piece.

Distance is a predefined delegate that assigns to pieces a heuristic value equaling the piece distance to the target. Thus of the four pieces adjacent to the current one, the piece closest to the target has the smallest value and the route generation continues through it. **Distance** is the default value of the RouteHeuristics property.

Example

The following sample demonstrates how to use this property:

C# Copy Code

```
diagram.RoutingOptions.RouteHeuristics = RoutingOptions.Distance;
```

Visual Basic Copy Code

```
Diagram.RoutingOptions.RouteHeuristics = RoutingOptions.Distance
```

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
DistSquare Property
MaxAbs Property

RoutingOptions.DistSquare Property

Gets a heuristic delegate function that returns the square distance dx^2 + dy^2 as a cost evaluation of the distance between two pieces of the routing grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static RouteHeuristics DistSquare { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property DistSquare As RouteHeuristics
```

Property Value

A predefined RouteHeuristics delegate instance.

Remarks

While generating a path, the routing algorithm evaluates pieces of the routing grid adjacent to the current one. It takes into account the cost for passing trough a piece and adds to that cost a heuristic value calculated by a RouteHeuristics delegate. So the smaller heuristic value a piece has, the greater the chance the found route to go through that piece.

DistSquare is a predefined delegate that assigns to pieces a heuristic value equaling the piece distance to the target raised to the second power. **DistSquare** can be assigned to the RouteHeuristics property. By returning larger values than the default heuristic

function Distance, it makes the prediction part of the routing algorithm have greater impact on the evaluated routes.That would lead to faster path-finding, but the real cost functions might have less impact and thus the quality of the found routes could be low.

Example

The following sample demonstrates how to use this property:

C# Copy Code

```
diagram.RoutingOptions.RouteHeuristics = RoutingOptions.DistSquare;
```

Visual Basic Copy Code

```
Diagram.RoutingOptions.RouteHeuristics = RoutingOptions.DistSquare
```

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
MaxAbs Property

RoutingOptions.DontOptimizeLongRoutes Property

Gets or sets a value indicating whether long routes don't have their number of segments minimized in order to save up processing time.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool DontOptimizeLongRoutes { get; set; }
```

Visual Basic Copy Code

```
Public Property DontOptimizeLongRoutes As Boolean
```

Property Value

true to disable route optimization for long links; otherwise, **false**. The default is **false**.

Remarks

Segment optimization done on routed links might need much processing time. The control allows disabling that optimization for long links. That would make the found routes have a "staircase" pattern when they run diagonally.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace

RoutingOptions.EndOrientation Property

Gets or sets the desired orientation of the last segments of routed links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Orientation EndOrientation { get; set; }
```

Visual Basic

```
Public Property EndOrientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default value is **Auto**.

Remarks

For some diagrams it might be desirable that first and last segments of links always have a specific direction. For example, in database ER diagrams links usually connect to table rows on the left- or right-hand sides of tables; in that case their end segments must be horizontal. **EndOrientation** specifies what orientation of links' last segments to be applied by the routing algorithm.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
StartOrientation Property

RoutingOptions.EndPointsOffset Property

Gets or sets the minimum number of routing grid cells the end segments of links should occupy before they are allowed to bend.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public int EndPointsOffset { get; set; }
```

Visual Basic

```
Public Property EndPointsOffset As Integer
```

Property Value

Number of routing grid cells the end segments of links should occupy before a bend is allowed. The default value is 0.

Remarks

Set this property to a value greater than 0 to ensure the first bend the link makes is not too close to the node boundaries.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
GridSize Property

RoutingOptions.GridSize Property

Gets or sets the size of the pieces of the routing grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double GridSize { get; set; }
```

Visual Basic

```
Public Property GridSize As Double
```

Property Value

A **double** value specifying the size of the pieces of the routing grid. The default is 4.

Remarks

A smaller size of routing grid pieces helps finding more precise paths, however since the number of pieces increases, there would be more time needed to find a path.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace

RoutingOptions.IgnoreAnchoring Property

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public bool IgnoreAnchoring { get; }
```

Visual Basic

```
Public ReadOnly Property IgnoreAnchoring As Boolean
```

Property Value

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace

RoutingOptions.LengthCost Property

Gets or sets a value added to the total cost of a route for each piece of the routing grid occupied by the route.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public byte LengthCost { get; set; }
```

Visual Basic

```
Public Property LengthCost As Byte
```

Property Value

A byte value specifying the length cost. The default is 2.

Remarks

This is a penalty value added to the total cost of an evaluated link route for each piece of the routing grid it occupies. The routing algorithm aims to find paths that have a minimal cost, so assigning a greater value to **LengthCost** leads to generating shorter paths.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
CrossingCost Property
NodeVicinityCost Property
TurnCost Property

RoutingOptions.MaxAbs Property

Gets a heuristic delegate function that returns max(dx, dy) as a cost evaluation of the distance between two pieces of the routing grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static RouteHeuristics MaxAbs { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property MaxAbs As RouteHeuristics
```

Property Value

A predefined RouteHeuristics delegate instance.

Remarks

While generating a path, the routing algorithm evaluates pieces of the routing grid adjacent to the current one. It takes into account the cost for passing trough a piece and adds to that cost a heuristic value calculated by a RouteHeuristics delegate function. So the smaller heuristic value a piece has, the greater the chance the found route to go through that piece.

MaxAbs is a predefined delegate that assigns to pieces the higher value from the horizontal or vertical -only distances between a piece and the target. **MaxAbs** can be assigned to the RouteHeuristics property. By returning smaller values than the default heuristic function Distance, it makes the prediction part of the routing algorithm have less impact on the evaluated routes. That might lead to slower path-finding, but the real cost functions would have greater impact and thus the quality of the found routes could be higher.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace

RoutingOptions.NodeVicinityCost Property

Gets or sets a value added to the total cost of a route if the route passes too close to a node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public byte NodeVicinityCost { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeVicinityCost As Byte
```

Property Value

A byte value specifying the node vicinity cost. The default is 2.

Remarks

This is a penalty value added to the total cost of an evaluated link route if it passes too close to a node. The routing algorithm aims to find paths that have a minimal cost, so assigning a greater value to **NodeVicinityCost** makes links avoid passing close to nodes. The size of the buffer area around nodes in which passing links get a penalty cost is set via NodeVicinitySize.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
CrossingCost Property
LengthCost Property
TurnCost Property

RoutingOptions.NodeVicinitySize Property

Gets or sets the size of a buffer zone around a node for which routes get a penalty cost if they pass through it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double NodeVicinitySize { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeVicinitySize As Double
```

Property Value

A **double** value specifying the size of the buffer area in which passing routes receive a penalty cost. The default is 12.

Remarks

The pieces of the routing grid that are closest to a node have the value of NodeVicinityCost assigned as their cost. The outer pieces of the buffer zone are assigned a portion of **NodeVicinityCost** proportional to their distance to the node.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace

RoutingOptions.PolylineRouting Property

Gets or sets a value indicating whether the segments of Polyline links should be oriented orthogonally to node's borders and each other.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PolylineRouting PolylineRouting { get; set; }
```

Visual Basic

```
Public Property PolylineRouting As PolylineRouting
```

Property Value

A member of the PolylineRouting enumeration. The default value is **AllOrthogonal**.

Remarks

This property is valid only for links whose Shape is set to Polyline.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace

RoutingOptions.RouteHeuristics Property

Gets or sets a heuristic function used to evaluate the distance between two pieces of the routing grid.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public RouteHeuristics RouteHeuristics { get; set; }
```

Visual Basic

```
Public Property RouteHeuristics As RouteHeuristics
```

Property Value

A RouteHeuristics delegate used to evaluate the distance from a current routing grid piece to the target one. The default is Distance.

Remarks

While generating a path, the routing algorithm evaluates pieces of the routing grid adjacent to the current one. It takes into account the cost for passing trough a piece and adds to that cost a heuristic value calculated by **RouteHeuristics**. So the smaller heuristic value a piece has, the greater the chance the found route to go through that piece.

There are three predefined heuristic functions that can be assigned to **RouteHeuristics**: Distance, DistSquare and MaxAbs. All of them assign to pieces heuristic values that are larger for larger distances to the target. Thus of the four pieces adjacent to the current one, the piece closest to the target has the smallest value and the route generation continues through it.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace

RoutingOptions.StartOrientation Property

Gets or sets the desired orientation of the first segments of routed links.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Orientation StartOrientation { get; set; }
```

Visual Basic Copy Code

```
Public Property StartOrientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default is **Auto**.

Remarks

For some diagrams it might be desirable that first and last segments of links always have a specific direction. For example, in database ER diagrams links usually connect to table rows on the left- or right-hand sides of tables; in that case their end segments must be horizontal. **StartOrientation** specifies what orientation of links' first segments to be applied by the routing algorithm.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
EndOrientation Property

RoutingOptions.TriggerRerouting Property

Gets or sets a value indicating under what circumstances links are rerouted automatically.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RerouteLinks TriggerRerouting { get; set; }
```

Visual Basic Copy Code

```
Public Property TriggerRerouting As RerouteLinks
```

Property Value

A member of the RerouteLinks enumeration. The default is **WhenModified**.

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace

RoutingOptions.TurnCost Property

Gets or sets a value added to the total cost of a route if it makes a turn.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public byte TurnCost { get; set; }
```

Visual Basic Copy Code

```
Public Property TurnCost As Byte
```

Property Value

A byte value specifying the turn cost. The default is 4.

Remarks

This is a penalty value added to the total cost of an evaluated link route for each turn it makes. The routing algorithm aims to find paths that have a minimal cost, so assigning a greater value to **TurnCost** leads paths that make less turns (hence they have less segments).

See Also

RoutingOptions Members
RoutingOptions Class
MindFusion.Diagramming Namespace
CrossingCost Property
LengthCost Property
NodeVicinityCost Property

RowEventArgs Class

Contains the arguments passed to handlers of table row-related events, such as TableSectionCollapsed and TableSectionExpanded.

Syntax

C# Copy Code

```
public class RowEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class RowEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.RowEventArgs

See Also

RowEventArgs Members
MindFusion.Diagramming Namespace

RowEventArgs Members

Contains the arguments passed to handlers of table row-related events, such as TableSectionCollapsed and TableSectionExpanded.

The following tables list the members exposed by the RowEventArgs type.

Public Constructors

	Name	Description
	RowEventArgs	Initializes a new instance of the **RowEventArgs** class.

Public Properties

	Name	Description
	RowIndex	Gets the index of the table row for which the event has been raised.
	Table	Get the table for which the event has been raised.

See Also

RowEventArgs Constructor

Initializes a new instance of the RowEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public RowEventArgs (
    TableNode table,
    int rowIndex
)
```

Visual Basic Copy Code

```
Public New ( _
    table As TableNode, _
    rowIndex As Integer _
)
```

Parameters

table
The TableNode related to the event.

rowIndex
The zero-based index of the table row related to the event.

See Also

RowEventArgs Class
MindFusion.Diagramming Namespace

RowEventArgs Properties

RowEventArgs.RowIndex Property

Gets the index of the table row for which the event has been raised.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int RowIndex { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property RowIndex As Integer
```

Property Value

An integer value specifying the row's index.

See Also

RowEventArgs Members
RowEventArgs Class

RowEventArgs.Table Property

Get the table for which the event has been raised.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode Table { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Table As TableNode
```

Property Value

An instance of the TableNode class.

See Also

RowEventArgs Members
RowEventArgs Class
MindFusion.Diagramming Namespace

SaveToStringFormat Enumeration

Specifies in what format to save the diagram when generating a string using the SaveToString method.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum SaveToStringFormat
```

Visual Basic Copy Code

```
Public Enum SaveToStringFormat
```

Members

	Member name	Description
	Xml	Indicates a non-compressed XML output.
	CompressedXml	Indicates a compressed XML output.
	Json	

See Also

MindFusion.Diagramming Namespace

ScriptHelper Class

Allows calling some static MvcDiagram methods from JavaScript and VBScript. It seems the script languages do not allow calling class static methods, so a ScriptHelper instance property is provided by the Diagram class.

Syntax

C# Copy Code

```
public class ScriptHelper
```

Visual Basic Copy Code

```
Public Class ScriptHelper
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ScriptHelper

See Also

ScriptHelper Members
MindFusion.Diagramming Namespace

ScriptHelper Members

Allows calling some static MvcDiagram methods from JavaScript and VBScript. It seems the script languages do not allow calling class static methods, so a ScriptHelper instance property is provided by the Diagram class.

The following tables list the members exposed by the ScriptHelper type.

Public Constructors

	Name	Description
	ScriptHelper	Initializes a new instance of the **ScriptHelper** class.

Public Methods

	Name	Description
	ApplyLayout	Applies the specified layout to the underlying diagram.
	CreateObject	Creates a MvcDiagram object given its class name.
	CreateSolidBrush	Creates a solid brush with the specified color components.
	NodesIntersect	Calculates whether the specified nodes' outlines intersect.
	ShapeTemplateFromId	Returns the Shape having the specified identifier.

See Also

ScriptHelper Class
MindFusion.Diagramming Namespace

ScriptHelper Constructor

Initializes a new instance of the ScriptHelper class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ScriptHelper (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram _
)
```

Parameters

diagram
The Diagram the helper methods will work on.

See Also

ScriptHelper Class
MindFusion.Diagramming Namespace

ScriptHelper Methods

ScriptHelper.ApplyLayout Method

Applies the specified layout to the underlying diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ApplyLayout (
    Object layout
)
```

Visual Basic Copy Code

```
Public Sub ApplyLayout( _
    layout As Object _
)
```

Parameters

layout
An instance of one of the layout classes that represents the layout to perform on the underlying diagram.

See Also

ScriptHelper Members
ScriptHelper Class
MindFusion.Diagramming Namespace

ScriptHelper.CreateObject Method

Creates a MvcDiagram object given its class name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object CreateObject (
    string className
)
```

Visual Basic Copy Code

```
Public Function CreateObject( _
    className As String _
) As Object
```

Parameters

className
The class name of the object to create.

Return Value

A reference to the MvcDiagram object of the specified class.

See Also

ScriptHelper Members
ScriptHelper Class
MindFusion.Diagramming Namespace

ScriptHelper.CreateSolidBrush Method

Creates a solid brush with the specified color components.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush CreateSolidBrush (
    int alpha,
    int red,
    int green,
    int blue
)
```

Visual Basic Copy Code

```
Public Function CreateSolidBrush( _
    alpha As Integer, _
    red As Integer, _
    green As Integer, _
    blue As Integer _
) As Brush
```

Parameters

alpha
The alpha factor of the solid brush color.

red
The red component of the solid brush color.

green
The green component of the solid brush color.

blue
The blue component of the solid brush color.

Return Value

An instance of the MindFusion.Drawing.SolidBrush that represents the newly created brush.

See Also

ScriptHelper Members
ScriptHelper Class
MindFusion.Diagramming Namespace

ScriptHelper.NodesIntersect Method

Calculates whether the specified nodes' outlines intersect.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool NodesIntersect (
    DiagramNode n1,
    DiagramNode n2
)
```

Visual Basic Copy Code

```
Public Function NodesIntersect( _
    n1 As DiagramNode, _
    n2 As DiagramNode _
) As Boolean
```

Parameters

n1
The first DiagramNode to check.

n2
The second **DiagramNode** to check.

Return Value

true if the outlines of the specified nodes intersect; otherwise, **false**.

See Also

ScriptHelper Members
ScriptHelper Class
MindFusion.Diagramming Namespace

ScriptHelper.ShapeTemplateFromId Method

Returns the Shape having the specified identifier.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape ShapeTemplateFromId (
    string shapeId
)
```

Visual Basic Copy Code

```
Public Function ShapeTemplateFromId( _
    shapeId As String _
) As Shape
```

Parameters

shapeId
The id of the shape to create.

Return Value

An instance of the Shape class.

See Also

ScriptHelper Members
ScriptHelper Class
MindFusion.Diagramming Namespace

Scrollable Interface

Interface used by scrollbars to interact with their containing nodes.

Syntax

C# Copy Code

```
public interface Scrollable
```

Visual Basic Copy Code

```
Public Interface Scrollable
```

Inheritors

MindFusion.Diagramming.ContainerNode
MindFusion.Diagramming.TableNode
MindFusion.Diagramming.TreeViewNode

See Also

Scrollable Members
MindFusion.Diagramming Namespace

Scrollable Members

Interface used by scrollbars to interact with their containing nodes.

The following tables list the members exposed by the Scrollable type.

Public Methods

	Name	Description
	CreateScrollCommand	Creates a Command -derived object that can be used to undo the scroll operation.
	GetScrollbarLargeChange	Gets the amount to scroll by when the user clicks above or below the scrollbar thumb.
	GetScrollbarMaximum	Gets the maximum scroll position.
	GetShape	Returns the node's shape.
	IntegralScrollMode	Checks if the scrollbar should report only integer values as scroll positions.
	UpdateScrollPosition	Updates the node's scroll position from scrollbar's value.

See Also

Scrollable Methods

Scrollable.CreateScrollCommand Method

Creates a Command -derived object that can be used to undo the scroll operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
Command CreateScrollCommand ()
```

Visual Basic Copy Code

```
Function CreateScrollCommand () As Command
```

Return Value

A **Command** object that can be used to undo the scroll operation. Target nodes return different commands depending on the type of their contents, e.g. ScrollContainerCmd for scrolling child nodes inside ContainerNode or ScrollTableCmd for scrolling rows of TableNode.

See Also

Scrollable Members
Scrollable Interface
MindFusion.Diagramming Namespace

Scrollable.GetScrollbarLargeChange Method

Gets the amount to scroll by when the user clicks above or below the scrollbar thumb.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
double GetScrollbarLargeChange (
    bool vertical
)
```

Visual Basic Copy Code

```
Function GetScrollbarLargeChange( _
    vertical As Boolean _
) As Double
```

Parameters

vertical
true if scrolling using a vertical scrollbar, or **false** if horizontal one.

Return Value

A **double** value specifying the amount to scroll.

See Also

Scrollable.GetScrollbarMaximum Method

Gets the maximum scroll position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
double GetScrollbarMaximum (
    bool vertical
)
```

Visual Basic Copy Code

```
Function GetScrollbarMaximum( _
    vertical As Boolean _
) As Double
```

Parameters

vertical
true if scrolling using a vertical scrollbar, or **false** if horizontal one.

Return Value

The maximum scroll position for specified scroll direction.

See Also

Scrollable Members
Scrollable Interface
MindFusion.Diagramming Namespace

Scrollable.GetShape Method

Returns the node's shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
SimpleShape GetShape ()
```

Visual Basic Copy Code

```
Function GetShape () As SimpleShape
```

Return Value

A member of the SimpleShape enumeration.

See Also

Scrollable Members
Scrollable Interface
MindFusion.Diagramming Namespace

Scrollable.IntegralScrollMode Method

Checks if the scrollbar should report only integer values as scroll positions.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
bool IntegralScrollMode ()
```

Visual Basic Copy Code

```
Function IntegralScrollMode () As Boolean
```

Return Value

true if the scroll target expects integer values for scroll offset, or **false** otherwise.

See Also

Scrollable Members
Scrollable Interface
MindFusion.Diagramming Namespace

Scrollable.UpdateScrollPosition Method

Updates the node's scroll position from scrollbar's value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
void UpdateScrollPosition (
    double newValue,
    bool vertical
)
```

Visual Basic Copy Code

```
Sub UpdateScrollPosition( _
    newValue As Double, _
    vertical As Boolean _
)
```

Parameters

newValue

A **double** value specifying scroll offset.

vertical

true if scrolling using a vertical scrollbar, or **false** if horizontal one.

See Also

Scrollable Members
Scrollable Interface
MindFusion.Diagramming Namespace

Selection Class

Represents a set of items selected in the diagram document.

Syntax

C# Copy Code

```
public class Selection : DiagramItem
```

Visual Basic Copy Code

```
Public Class Selection
    Inherits DiagramItem
```

Remarks

In a diagram document, several items can be selected simultaneously. If more than one item is selected, the compound set of selected items is handled by an instance of the **Selection** class.

Items in the selection can be of any element type. The Items property gives a collection of all the items in the selection. You can get type-safe access to items of specific class by using the Links and Nodes collections properties.

Users can select multiple items by drawing a selection rectangle. Depending on the **DiagramView**'s **Behavior**, this is done either by simply dragging the mouse, or by holding CTRL down while dragging the mouse. Depending on the IncludeItemsIfIntersect value, items are selected if they are entirely inside the drawn selection rectangle, or if at least a part of them is inside the rectangle. It is possible to prevent users from selecting more than one items at the same time by setting AllowMultipleSelection to **false**.

Items can be always added or removed programmatically to selection by calling the AddItem and RemoveItem methods. Selection state of an item can be toggled, getting deselected from selected and vice versa, by calling the Toggle method. All items in the selection are deselected if Clear is invoked.

Use the Selected property of the DiagramItem class to determine whether a particular item is selected. The Selection's GetSize method returns how many items are selected. Bounds returns the rectangle that bounds all selected items.

The visual style of multiple selection is defined by the Style property. For example selection might be painted as a dashed rectangle or a semi-transparent filled rectangle.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.Selection

See Also

Selection Members
MindFusion.Diagramming Namespace

Selection Members

Represents a set of items selected in the diagram document.

The following tables list the members exposed by the Selection type.

Public Constructors

	Name	Description
	Selection	Initializes a new instance of the **Selection** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowMultipleSelection	Gets or sets a value indicating whether more than one items can be selected simultaneously.

	Bounds	Gets the rectangle that bounds all selected items.
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	ExtendMode	Gets or sets whether to prevent deselecting currently selected items when they are outside of a newly drawn lasso rectangle.
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncludeItemsIfIntersect	Gets or sets a value indicating which items are selected when drawing a selection rectangle.
	Items	Gets a collection of all selected items.
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Links	Gets the collection of all links in the selection.
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	Nodes	Gets a collection of all selected nodes.
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	SetsDirtyFlag	Gets or sets a value indicating whether the diagram's Dirty flag should be set when selection changes.
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)

	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets a value indicating how to display multiple selection of items.
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	True if this is a top-level item in the diagram, or false if it is inside a container. (Inherited from DiagramItem.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AddItem	Adds a new item to the current selection.
	Animate	Animates the item. (Inherited from DiagramItem.)
	Change	Selects an item and deselects all others.
	Clear	Removes all items from the selection.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	ContainsPoint	Overloaded.
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override.
	DrawHandles	DiagramItem.DrawHandles override.
	DrawInRect	(Inherited from DiagramItem.)

	DrawShadow	DiagramItem.DrawShadow override.
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetOutlinePoly	Gets a polygon approximating the outline of this object. (Inherited from DiagramItem.)
	GetRepaintRect	DiagramItem.GetRepaintRect override.
	GetRotatedBounds	Gets the bounding rectangle of this item when it is rotated. (Inherited from DiagramItem.)
	GetSize	Gets the number of selected items.
	GetToolTip	Gets the tooltip that should be displayed for this item at the specified point. (Inherited from DiagramItem.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override.
	HitTestManipulators	Finds the **Manipulator** object located at the specified point. (Inherited from DiagramItem.)
	LabelFromPoint	Returns the ItemLabel located at specified point. (Inherited from DiagramItem.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	RemoveItem	Removes an item from the selection.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	Toggle	Toggles the selection state of an item.
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description

	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override.
	AllowModify	DiagramItem.AllowModify override.
	CancelModify	DiagramItem.CancelModify override.
	CompleteCreate	DiagramItem.CompleteCreate override.
	CompleteModify	DiagramItem.CompleteModify override.
	CreateProperties	Creates an item property set corresponding to the type of this diagram item. (Inherited from DiagramItem.)
	CreateState	Creates an item state corresponding to the type of this diagram item. (Inherited from DiagramItem.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	DiagramItem.FreeResources override.
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	LoadFromJson	Loads the item from a JSON object. (Inherited from DiagramItem.)
	LoadFromXml	Loads the item content from an XML element. (Inherited from DiagramItem.)
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	Called when the item is added to a Diagram, either by the user, programmatically or via undo / redo. (Inherited from DiagramItem.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	Called when the diagram's MeasureUnit changes. (Inherited from DiagramItem.)
	OnRemove	Called when an item is removed from a **Diagram**, either by the user, programmatically or via undo / redo. (Inherited from DiagramItem.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnZoomChanged	Called when the diagram's **ZoomFactor** changes. (Inherited from DiagramItem.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	Restores the properties of this diagram item. (Inherited from DiagramItem.)
	RestoreState	Restores the state of this diagram item. (Inherited from DiagramItem.)

	SaveProperties	Saves the properties of this diagram item. (Inherited from DiagramItem.)
	SaveState	Saves the state of this diagram item. (Inherited from DiagramItem.)
	SaveToJson	Saves the item as a JSON object. (Inherited from DiagramItem.)
	SaveToXml	Saves the item content into an XML element. (Inherited from DiagramItem.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpanded	Sets whether the item is in expanded or collapsed state. (Inherited from DiagramItem.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	StartCreate	DiagramItem.StartCreate override.
	StartModify	DiagramItem.StartModify override.
	UpdateCreate	Invoked during interactive creation of this item. (Inherited from DiagramItem.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateModify	DiagramItem.UpdateModify override.

See Also

Selection Class
MindFusion.Diagramming Namespace

Selection Constructor

Initializes a new instance of the Selection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Selection (
    Diagram parent,
    bool toolMode,
    IDiagramView zoomTarget
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram, _
    toolMode As Boolean, _
    zoomTarget As IDiagramView _
)
```

Parameters

parent
The Diagram instance.

toolMode
true if this object is used only to draw lasso, or **false** if used to select diagram items.

zoomTarget
The diagram view zoomed by lasso tool.

See Also

Selection Class
MindFusion.Diagramming Namespace

Selection Methods

Selection.AddItem Method

Adds a new item to the current selection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void AddItem (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Sub AddItem( _
    item As DiagramItem _
)
```

Parameters

item
The DiagramItem to be added to the selection.

Remarks

This method adds a new item to the selection. It also deactivates the ActiveItem, if any, and makes the newly added item the active one.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace
Change Method
RemoveItem Method
Clear Method

Selection.AllowCreate Method

DiagramItem.AllowCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override bool AllowCreate (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Function AllowCreate( _
    current As Point, _
    ist As InteractionState _
) As Boolean
```

Parameters

current

ist

Return Value

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.AllowModify Method

DiagramItem.AllowModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override bool AllowModify (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Function AllowModify( _
    current As Point, _
    ist As InteractionState _
) As Boolean
```

Parameters

current

ist

Return Value

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.CancelModify Method

DiagramItem.CancelModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void CancelModify (
    InteractionState ist
```

```
)
```

Visual Basic Copy Code

```
Protected Overrides Sub CancelModify( _
    ist As InteractionState _
)
```

Parameters

ist

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.Change Method

Selects an item and deselects all others.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Change (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Sub Change( _
    item As DiagramItem _
)
```

Parameters

item

The DiagramItem which should be made active.

Remarks

Call this method to select an item and make it the ActiveItem. The method doesn't let you select locked items.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace
AddItem Method
RemoveItem Method
Clear Method

Selection.Clear Method

Removes all items from the selection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Clear ()
```

Visual Basic Copy Code

```
Public Sub Clear ()
```

Remarks

Call **Clear** to deselect all items. The items are not deleted. The method also sets the diagram's ActiveItem to **null** (**Nothing** in Visual Basic).

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace
AddItem Method
RemoveItem Method

Selection.CompleteCreate Method

DiagramItem.CompleteCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void CompleteCreate (
    Point end
)
```

Visual Basic Copy Code

```
Protected Overrides Sub CompleteCreate( _
    end As Point _
)
```

Parameters

end

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.CompleteModify Method

DiagramItem.CompleteModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void CompleteModify (
    Point end,
    InteractionState ist
)
```

Visual Basic

```
Protected Overrides Sub CompleteModify( _
    end As Point, _
    ist As InteractionState _
)
```

Parameters

end

ist

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.ContainsPoint Method

Overload List

Name	Description
Selection.ContainsPoint (Point)	DiagramItem.ContainsPoint override.
Selection.ContainsPoint (Point, Double)	DiagramItem.ContainsPoint override.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.ContainsPoint Method (Point)

DiagramItem.ContainsPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override bool ContainsPoint (
    Point point
)
```

Visual Basic

```
Public Overrides Function ContainsPoint( _
    point As Point _
) As Boolean
```

Parameters

point
The point to check.

Return Value

See Also

ContainsPoint Method Overload List
Selection Members

Selection.ContainsPoint Method (Point, Double)

DiagramItem.ContainsPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool ContainsPoint (
    Point point,
    double threshold
)
```

Visual Basic Copy Code

```
Public Overrides Function ContainsPoint( _
    point As Point, _
    threshold As Double _
) As Boolean
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying the logical coordinates of a diagram point.

threshold
Specifies the size of the area around the item outlines where the point is still considered inside the item.

Return Value

true it this item contains the specified point; otherwise, **false**.

See Also

ContainsPoint Method Overload List
Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.Draw Method

DiagramItem.Draw override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void Draw (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub Draw( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics

options

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.DrawHandles Method

DiagramItem.DrawHandles override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawHandles (
    IGraphics graphics,
    HandlesVisualStyle style1,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawHandles( _
    graphics As IGraphics, _
    style1 As HandlesVisualStyle, _
    options As RenderOptions _
)
```

Parameters

graphics

style1

options

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.DrawShadow Method

DiagramItem.DrawShadow override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawShadow (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawShadow( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics

An IGraphics object where the shadow should be rendered.

options

A RenderOptions object specifying options for the drawing operation.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.FreeResources Method

DiagramItem.FreeResources override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void FreeResources ()
```

Visual Basic Copy Code

```
Protected Overrides Sub FreeResources ()
```

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.GetRepaintRect Method

DiagramItem.GetRepaintRect override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Rect GetRepaintRect (
    bool connected
)
```

Visual Basic Copy Code

```
Public Overrides Function GetRepaintRect( _
    connected As Boolean _
) As Rect
```

Parameters

connected

Return Value

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.GetSize Method

Gets the number of selected items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int GetSize ()
```

Visual Basic Copy Code

```
Public Function GetSize () As Integer
```

Return Value

An integer value indicating the size of the collection.

Remarks

Call this method to get the number of selected items, including both nodes and links.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace
Items Property

Selection.HitTestHandle Method

DiagramItem.HitTestHandle override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override AdjustmentHandle HitTestHandle (
    Point point,
    double distance
)
```

Visual Basic Copy Code

```
Public Overrides Function HitTestHandle( _
    point As Point, _
    distance As Double _
) As AdjustmentHandle
```

Parameters

point

distance

Return Value

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.RemoveItem Method

Removes an item from the selection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RemoveItem (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Function RemoveItem( _
    item As DiagramItem _
) As Boolean
```

Parameters

item

The DiagramItem to be removed from the selection.

Return Value

true if the item has been successfully removed; otherwise, **false**.

Remarks

Call this method to deselect an item. Note that the item is not deleted, but just removed from the selection.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace
AddItem Method

Selection.StartCreate Method

DiagramItem.StartCreate override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void StartCreate (
    Point org
)
```

Visual Basic Copy Code

```
Protected Overrides Sub StartCreate( _
    org As Point _
```

```
)
```

Parameters

org

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.StartModify Method

DiagramItem.StartModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void StartModify (
    Point org,
    AdjustmentHandle handle,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub StartModify( _
    org As Point, _
    handle As AdjustmentHandle, _
    ist As InteractionState _
)
```

Parameters

org

handle

ist

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.Toggle Method

Toggles the selection state of an item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Toggle (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Sub Toggle( _
    item As DiagramItem _
)
```

Parameters

item
The DiagramItem whose selection state should be changed.

Remarks

Call this method to toggle an item's selection state. The method selects the diagram item passed as argument if previously it was deselected. If the item was selected, the method deselects it.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace
Change Method
AddItem Method
RemoveItem Method

Selection.UpdateModify Method

DiagramItem.UpdateModify override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void UpdateModify (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub UpdateModify( _
    current As Point, _
    ist As InteractionState _
)
```

Parameters

current

ist

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection Properties

Selection.AllowMultipleSelection Property

Gets or sets a value indicating whether more than one items can be selected simultaneously.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowMultipleSelection { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowMultipleSelection As Boolean
```

Property Value

A boolean flag indicating whether multiple items can be selected simultaneously by the user.

Remarks

The property controls whether multiple items can be selected interactively. If it is disabled, multiple items can still be selected programmatically.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.Bounds Property

Gets the rectangle that bounds all selected items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect Bounds { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Bounds As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** instance.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.ExtendMode Property

Gets or sets whether to prevent deselecting currently selected items when they are outside of a newly drawn lasso rectangle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ExtendMode { get; set; }
```

Visual Basic Copy Code

```
Public Property ExtendMode As Boolean
```

Property Value

false to deselect items located out of newly drawn lasso rectangle, or **true** to keep them selected. The default value is **false**.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.IncludeItemsIfIntersect Property

Gets or sets a value indicating which items are selected when drawing a selection rectangle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IncludeItemsIfIntersect { get; set; }
```

Visual Basic Copy Code

```
Public Property IncludeItemsIfIntersect As Boolean
```

Property Value

true to select items that intersect the drawn selection rectangle; **false** to select items only if they are entirely inside the selection rectangle. The default is **true**.

Remarks

The value of this property is considered when users draw a selection rectangle using a pointing device. By default, all items inside the drawn rectangle and the ones that are only partially inside it are selected. Set the property to **false** to let users select items only when the items are entirely inside the selection rectangle.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.Items Property

Gets a collection of all selected items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItemCollection Items { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Items As DiagramItemCollection
```

Property Value

An instance of the DiagramItemCollection class that represents the selected diagram items.

Remarks

You can select items programmatically by adding them to this collection.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace
Nodes Property
Links Property

Selection.Links Property

Gets the collection of all links in the selection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection Links { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Links As DiagramLinkCollection
```

Property Value

An instance of the DiagramLinkCollection class that represents the selected diagram items.

Remarks

You can select links programmatically by adding them to this collection.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace
Items Property
Nodes Property

Selection.Nodes Property

Gets a collection of all selected nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeCollection Nodes { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Nodes As DiagramNodeCollection
```

Property Value

An instance of the DiagramNodeCollection class that represents the nodes in the selection.

Remarks

You can select nodes programmatically by adding them to this collection.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace
Items Property
Links Property

Selection.SetsDirtyFlag Property

Gets or sets a value indicating whether the diagram's Dirty flag should be set when selection changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool SetsDirtyFlag { get; set; }
```

Visual Basic Copy Code

```
Public Property SetsDirtyFlag As Boolean
```

Property Value

true to set the flag when selection changes, otherwise **false**. The default value is **true**.

See Also

Selection Members
Selection Class
MindFusion.Diagramming Namespace

Selection.Style Property

Gets or sets a value indicating how to display multiple selection of items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public new SelectionStyle Style { get; set; }
```

Visual Basic Copy Code

```
Public Shadows Property Style As SelectionStyle
```

Property Value

A member of the SelectionStyle enumeration. The default is **SelectionHandles**.

Remarks

The property specifies how multiple selection is presented visually to users - with manipulation handles painted around each item or with a rectangle painted around all items.

See Also

SelectionCopy Class

Used internally in the implementation of copy/paste functions.

Syntax

C# Copy Code

```
public sealed class SelectionCopy
```

Visual Basic Copy Code

```
Public NotInheritable Class SelectionCopy
```

Remarks

This class supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.SelectionCopy

See Also

SelectionCopy Members
MindFusion.Diagramming Namespace

SelectionCopy Members

Used internally in the implementation of copy/paste functions.

The following tables list the members exposed by the SelectionCopy type.

Public Constructors

	Name	Description
	SelectionCopy	Initializes a new instance of the **SelectionCopy** class.

Public Properties

	Name	Description
	Groups	Gets the underlying collection of groups.
	Items	Gets the underlying collection of items.
	OriginalItems	Gets a collection of the original items.

See Also

SelectionCopy Class
MindFusion.Diagramming Namespace

SelectionCopy Constructor

Initializes a new instance of the SelectionCopy class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SelectionCopy (
    DiagramItemCollection items,
    DiagramItemCollection originalItems,
    GroupCollection groups
)
```

Visual Basic Copy Code

```
Public New ( _
    items As DiagramItemCollection, _
    originalItems As DiagramItemCollection, _
    groups As GroupCollection _
)
```

Parameters

items
A collection containing the item copies.

originalItems
A collection containing the original items.

groups
A collection containing the group copies.

Remarks

This constructor supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

SelectionCopy Class
MindFusion.Diagramming Namespace

SelectionCopy Properties

SelectionCopy.Groups Property

Gets the underlying collection of groups.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public GroupCollection Groups { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Groups As GroupCollection
```

Property Value

An instance of the GroupCollection class.

Remarks

This property supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

SelectionCopy Members
SelectionCopy Class
MindFusion.Diagramming Namespace

SelectionCopy.Items Property

Gets the underlying collection of items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItemCollection Items { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Items As DiagramItemCollection
```

Property Value

An instance of the DiagramItemCollection class.

Remarks

This property supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

SelectionCopy Members
SelectionCopy Class
MindFusion.Diagramming Namespace

SelectionCopy.OriginalItems Property

Gets a collection of the original items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItemCollection OriginalItems { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property OriginalItems As DiagramItemCollection
```

Property Value

An instance of the DiagramItemCollection class.

Remarks

This property supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

SelectionCopy Members
SelectionCopy Class
MindFusion.Diagramming Namespace

SelectionEventArgs Class

Contains the arguments passed to the SelectionModified event handler.

Syntax

C# Copy Code

```
public class SelectionEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class SelectionEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.SelectionEventArgs
      MindFusion.Diagramming.SelectionValidationEventArgs

See Also

SelectionEventArgs Members
MindFusion.Diagramming Namespace

SelectionEventArgs Members

Contains the arguments passed to the SelectionModified event handler.

The following tables list the members exposed by the SelectionEventArgs type.

Public Constructors

	Name	Description
	SelectionEventArgs	Initializes a new instance of the SelectionEventArgs class.

Public Properties

	Name	Description
	AdjustmentHandle	Gets the adjustment handle being moved by the user.
	MousePosition	Gets the current mouse pointer position.
	Selection	Gets the Selection object representing current selection lasso tool.

See Also

SelectionEventArgs Class
MindFusion.Diagramming Namespace

SelectionEventArgs Constructor

Initializes a new instance of the SelectionEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SelectionEventArgs (
    Point mousePosition,
    AdjustmentHandle adjustmentHandle
)
```

Visual Basic Copy Code

```
Public New ( _
    mousePosition As Point, _
    adjustmentHandle As AdjustmentHandle _
)
```

Parameters

mousePosition

adjustmentHandle

See Also

SelectionEventArgs Class
MindFusion.Diagramming Namespace

SelectionEventArgs Properties

SelectionEventArgs.AdjustmentHandle Property

Gets the adjustment handle being moved by the user.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AdjustmentHandle AdjustmentHandle { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property AdjustmentHandle As AdjustmentHandle
```

Property Value

See Also

SelectionEventArgs Members
SelectionEventArgs Class
MindFusion.Diagramming Namespace

SelectionEventArgs.MousePosition Property

Gets the current mouse pointer position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point MousePosition { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MousePosition As Point
```

Property Value

See Also

SelectionEventArgs Members
SelectionEventArgs Class
MindFusion.Diagramming Namespace

SelectionEventArgs.Selection Property

Gets the Selection object representing current selection lasso tool.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Selection Selection { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Selection As Selection
```

Property Value

A Selection object implementing selection lasso tool.

See Also

SelectionEventArgs Members
SelectionEventArgs Class
MindFusion.Diagramming Namespace

SelectionStyle Enumeration

Defines values that specify how multiple selected items are indicated visually.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum SelectionStyle
```

Visual Basic Copy Code

```
Public Enum SelectionStyle
```

Members

	Member name	Description
	SelectionHandles	Rectangular manipulation handles are painted over selected items.
	Frame	A dashed rectangle is drawn around selected items.

Remarks

These values affect multiple selection of items. If only a single item is selected, it is drawn with manipulation handles around it.

See Also

MindFusion.Diagramming Namespace

SelectionValidationEventArgs Class

Contains the arguments passed to the SelectionModifying event handler.

Syntax

C# Copy Code

```
public class SelectionValidationEventArgs : SelectionEventArgs
```

Visual Basic Copy Code

```
Public Class SelectionValidationEventArgs
    Inherits SelectionEventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.SelectionEventArgs
      MindFusion.Diagramming.SelectionValidationEventArgs

See Also

SelectionValidationEventArgs Members
MindFusion.Diagramming Namespace

SelectionValidationEventArgs Members

Contains the arguments passed to the SelectionModifying event handler.

The following tables list the members exposed by the SelectionValidationEventArgs type.

Public Constructors

	Name	Description
	SelectionValidationEventArgs	Initializes a new instance of the SelectionValidationEventArgs class.

Public Properties

	Name	Description
	AdjustmentHandle	Gets the adjustment handle being moved by the user. (Inherited from SelectionEventArgs.)
	Cancel	Gets or sets a value indicating whether to prevent the user from completing the operation.
	MousePosition	Gets the current mouse pointer position. (Inherited from SelectionEventArgs.)
	Selection	Gets the Selection object representing current selection lasso tool. (Inherited from SelectionEventArgs.)

Public Methods

	Name	Description
	CancelDrag	Cancels the modification of the selected items.

See Also

SelectionValidationEventArgs Class
MindFusion.Diagramming Namespace

SelectionValidationEventArgs Constructor

Initializes a new instance of the SelectionValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SelectionValidationEventArgs (
    Point mousePosition,
    AdjustmentHandle adjustmentHandle
)
```

Visual Basic Copy Code

```
Public New ( _
    mousePosition As Point, _
    adjustmentHandle As AdjustmentHandle _
)
```

Parameters

mousePosition

adjustmentHandle

See Also

SelectionValidationEventArgs Class
MindFusion.Diagramming Namespace

SelectionValidationEventArgs Methods

SelectionValidationEventArgs.CancelDrag Method

Cancels the modification of the selected items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void CancelDrag ()
```

Visual Basic Copy Code

```
Public Sub CancelDrag ()
```

See Also

SelectionValidationEventArgs Members
SelectionValidationEventArgs Class
MindFusion.Diagramming Namespace

SelectionValidationEventArgs Properties

SelectionValidationEventArgs.Cancel Property

Gets or sets a value indicating whether to prevent the user from completing the operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Cancel { get; set; }
```

Visual Basic Copy Code

```
Public Property Cancel As Boolean
```

Property Value

See Also

SelectionValidationEventArgs Members
SelectionValidationEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs Class

Contains the arguments passed to the SerializeTag and DeserializeTag event handlers.

Syntax

C# Copy Code

```
public class SerializeTagEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class SerializeTagEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.SerializeTagEventArgs

See Also

SerializeTagEventArgs Members
MindFusion.Diagramming Namespace

SerializeTagEventArgs Members

Contains the arguments passed to the SerializeTag and DeserializeTag event handlers.

The following tables list the members exposed by the SerializeTagEventArgs type.

Public Properties

	Name	Description

	Context	Gets a context object that can help you write and read various values.
	Handled	Gets or sets a value indicating whether the event is handled.
	Object	Gets the diagram element whose tag is being serialized/deserialized.
	PropertyName	Gets the name of the property being serialized.
	Representation	Gets or sets an **XmlElement** object where the tag should be serialized/deserialized.
	Tag	Gets or sets the tag object being serialized/deserialized.

See Also

SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs Properties

SerializeTagEventArgs.Context Property

Gets a context object that can help you write and read various values.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlPersistContext Context { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Context As XmlPersistContext
```

Property Value

An instance of the XmlPersistContext class.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs.Handled Property

Gets or sets a value indicating whether the event is handled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Handled { get; set; }
```

Visual Basic Copy Code

```
Public Property Handled As Boolean
```

Property Value

true if you have serialized/deserialized the tag object; otherwise, **false**.

Remarks

Always set this flag to **true** if you perform custom tag serialization.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs.Object Property

Gets the diagram element whose tag is being serialized/deserialized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Object { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Object As Object
```

Property Value

A diagram item, group, cell or anchor point.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs.PropertyName Property

Gets the name of the property being serialized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string PropertyName { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property PropertyName As String
```

Property Value

Either "Id" or "Tag", indicating which property is being serialized.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs.Representation Property

Gets or sets an **XmlElement** object where the tag should be serialized/deserialized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement Representation { get; set; }
```

Visual Basic Copy Code

```
Public Property Representation As XmlElement
```

Property Value

An instance of the .NET **XmlElement** class that represents the root of the XML branch where the tag is / should be stored.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

SerializeTagEventArgs.Tag Property

Gets or sets the tag object being serialized/deserialized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Tag { get; set; }
```

Visual Basic Copy Code

```
Public Property Tag As Object
```

Property Value

An object containing the custom tag data.

See Also

SerializeTagEventArgs Members
SerializeTagEventArgs Class
MindFusion.Diagramming Namespace

ShadowsStyle Enumeration

Enumerates possible shadows-drawing styles.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum ShadowsStyle
```

Visual Basic

```
Public Enum ShadowsStyle
```

Members

	Member name	Description
	None	No shadows are drawn.
	OneLevel	All shadows are drawn at the lowest z-level, and appear to be on one level.
	ZOrder	Every shadows is drawn just below its item, possibly covering other items.

Remarks

This enumeration provides the possible values for the ShadowsStyle property of the Diagram class.

See Also

MindFusion.Diagramming Namespace
ShadowsStyle Property

Shape Class

An instance of this class represents a shape defined through a series of arc, line and Bézier segments. The segments can be a part of the shape outline, in which case they define the part of the shape is filled, used for hit-testing and for aligning links to the node edges. Segments can also be added as decoration elements, in which case they only affect the appearance of a ShapeNode.

Syntax

C#

```
public class Shape : ICloneable
```

Visual Basic

```
Public Class Shape
    Implements ICloneable
```

Remarks

Shape Definitions

The **Shape** class provides the means for defining complex node shapes. With its help you can design any shape composed of lines, arcs and Bézier-curves. A shape template should contain an outline to be used for hit testing, clipping and finding intersections with other items. If the outline is not defined, the rectangle specified by the **Bounds** property of nodes is used for hit-testing. Optionally, shapes can contain decoration elements and text region definitions. Decorations are visual elements that do not take part in clipping, and are used for hit-testing only if DecorationHitTesting is enabled. Text regions are the parts of shapes in which node's text is laid out and rendered.

Predefined Shapes

MvcDiagram provides a set of about 100 predefined shape templates accessible through the static Shapes collection and the FromId method of **Shape**. A shape template can be applied to a node by assigning it to the Shape property of shape nodes. The predefined shapes are also exposed as properties of the Shapes class.

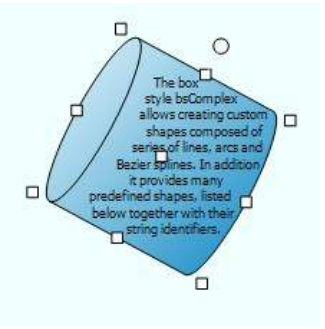
Custom Shapes

There are several Shape constructor overrides you might choose from, depending on the complexity of the shapes you want to create. They take arrays of ElementTemplate objects as arguments, which define the outline, decorations and text region definitions. Templates are described with coordinates expressed as percentage of a shape node's area. It is also possible to associate an image with a shape definition by means of the Image property and to define the area in which the image is displayed by ImageRectangle.

Dynamic Shapes

Shapes defined via ElementTemplate objects are scaled proportionally to the size of nodes. If shape elements should scale non-uniformly, use the Shape(formula) constructor to define shapes through Visio-like formulas that take into consideration the current width and height of nodes. Another constructor takes a CreatePathDelegate parameter that can be used to define shapes via .NET functions.

A formula shape can be parameterized by adding ShapeControlPoint objects to the shape's ControlPoints collection. The control point positions are passed as arguments to the shape scripts, where they can be used to modify the node's appearance. For example, the following code defines a rounded rectangle shape, whose corner radius can be modified by users via the control point.



C# Copy Code

```
// a rounded rectangle shape, with an arc at each corner
string roundRect = @"
    r = Min(Width / 2, radius.X);
    MoveTo(r, 0);
    LineTo(Width - r, 0);
    ArcTo(Width, r, false, false, r, r);
    LineTo(Width, Height - r);
    ArcTo(Width - r, Height, false, false, r, r);
    LineTo(r, Height);
    ArcTo(0, Height - r, false, false, r, r);
    LineTo(0, r);
    ArcTo(r, 0, false, false, r, r);
    ";

var myRect = new Shape(roundRect, "MyRect");

// add a control point for the 'radius' parameter
myRect.ControlPoints.Add(new ShapeControlPoint(
    "radius", 5, 1, 15, UnitType.Fixed, 0, 0, 0, UnitType.Fixed));
```

Shape Libraries

The **ShapeDesigner** tool included in the MvcDiagram suite lets you draw custom shapes and store them in shape libraries. A library file can be loaded into your application using the ShapeLibrary class. The definitions loaded from a shape library are automatically added to the Shapes collection and can be accessed through the FromId method, just as the predefined shapes.

Link Arrowheads

A **Shape** instance can be assigned to the HeadShape, BaseShape or IntermediateShape properties of a DiagramLink. When used for arrowheads, position (50, 0) in the shape definition corresponds to the arrowhead's tip point. Several predefined arrowhead shapes are provided as static properties of the ArrowHeads class.

Example

Here is how the DirectAccessStorage shape is defined in the MvcDiagram source code:

C# Copy Code

```
new Shape(

    // Outlines
    new ElementTemplate[]
    {
        new LineTemplate(10, 0, 90, 0),
        new ArcTemplate(80, 0, 20, 100, -90, 180),
        new LineTemplate(90, 100, 10, 100),
        new ArcTemplate(0, 0, 20, 100, 90, 180)
    },

    // Decorations
    new ElementTemplate[]
    {
        new ArcTemplate(0, 0, 20, 100, -90, 180)
    },

    // Text area
    new ElementTemplate[]
    {
        new LineTemplate(10, 0, 90, 0),
        new ArcTemplate(80, 0, 20, 100, -90, 180),
```

```
        new LineTemplate(90, 100, 10, 100),
        new BezierTemplate(10, 100, 35, 66, 35, 33, 10, 0)
    },

    FillMode.Winding, "DirectAccessStorage");
```

Visual Basic Copy Code

```
New Shape( _
    New ElementTemplate() _
    { _
        New LineTemplate(10, 0, 90, 0), _
        New ArcTemplate(80, 0, 20, 100, -90, 180), _
        New LineTemplate(90, 100, 10, 100), _
        New ArcTemplate(0, 0, 20, 100, 90, 180) _
    }, _
    New ElementTemplate() _
    { _
        New ArcTemplate(0, 0, 20, 100, -90, 180) _
    }, _
    New ElementTemplate() _
    { _
        New LineTemplate(10, 0, 90, 0), _
        New ArcTemplate(80, 0, 20, 100, -90, 180), _
        New LineTemplate(90, 100, 10, 100), _
        New BezierTemplate(10, 100, 35, 66, 35, 33, 10, 0) _
    }, _
    FillMode.Winding, "DirectAccessStorage")
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Shape

See Also

Shape Members
MindFusion.Diagramming Namespace

Shape Members

An instance of this class represents a shape defined through a series of arc, line and Bézier segments. The segments can be a part of the shape outline, in which case they define the part of the shape is filled, used for hit-testing and for aligning links to the node edges. Segments can also be added as decoration elements, in which case they only affect the appearance of a ShapeNode.

The following tables list the members exposed by the Shape type.

Public Constructors

	Name	Description
	Shape	Overloaded. Initializes a new instance of the **Shape** class.

Public Properties

	Name	Description
	AnchorPattern	Gets or sets the default anchor pattern for nodes that use this shape.
	ControlPoints	A list of ShapeControlPoint objects that let users adjust a node's shape.
	DecorationHitTesting	Gets or sets a value indicating whether decoration lines should be considered when hit-testing.
	Decorations	Gets or sets a list with all decorations of this shape.
	DecorationsFormula	Gets or sets a formula describing the shape decorations depending on the current width and height of the node.

	DisplayName	Specifies the name that should be displayed for this shape in user interface objects, for example the **ShapeListBox** or **ShapeToolBar**.
	EnableOutline	Gets or sets a value indicating whether to enable the shape's outline.
	FillMode	Gets or sets a value indicating how areas are filled if the outline path of the shape intersects itself.
	Id	Gets or sets the shape identifier.
	Image	Gets or sets the image displayed in all instances of this shape.
	ImageLocation	Gets or sets the path to the file that contains the image displayed in this shape.
	ImageRectangle	Gets or sets the position of the images displayed in instances of this shape.
	IsArrowhead	Specifies if this shape is used as arrowhead.
	LinkSegmentInset	Specifies what length of link's line segment is drawn inside the arrowhead shape, expressed as percentage of arrowhead size.
	Outline	Gets or sets the outline of this shape.
	OutlineFormula	Gets or sets a formula describing the shape outline depending on the current width and height of the node.
	ShapeDecorations	Gets or sets the ShapeDecoration objects associated with this shape.
S	Shapes	Gets a read-only collection of all named shapes.
	TextArea	Gets or sets the outline of the text area of this shape.

Public Methods

	Name	Description
	Clone	Creates an exact copy of this shape.
	Draw	Overloaded. Draws this shape on an IGraphics surface.
	Equals	Checks whether this shape template is equivalent to the specified shape.
S	FromId	Returns a reference to the shape with the specified string identifier.
S	FromSvg	Overloaded. Creates a **Shape** from the outline of an SVG drawing.
	GetHitPath	Returns a **GraphicsPath** object corresponding to the outline of the specified shape when rotated at the specified angle. That **GraphicsPath** can be used for hit-testing and intersection checks.
	GetRenderingPath	Returns a **GraphicsPath** object corresponding to the outline of the shape when rotated at the specified angle. That **GraphicsPath** can be used when drawing the shape's outline.
	LoadFromXml	Loads the shape content from an XML element.
	SaveToXml	Saves the shape content into an XML element.
	ToString	Returns the identifier of this shape template.

See Also

Shape Class
MindFusion.Diagramming Namespace

Shape Constructor

Initializes a new instance of the Shape class.

Overload List

Name	Description
Shape (Byte[])	Initializes a new instance of the **Shape** class with the specified outline.
Shape (Byte[], String)	Initializes a new instance of the **Shape** class with the specified outline and id.
Shape (CreatePathDelegate, CreatePathDelegate, String)	Initializes a new instance of the **Shape** class with the specified outline and decorations callbacks and id.
Shape (CreatePathDelegate, String)	Initializes a new instance of the **Shape** class with the specified outline delegate and id.
Shape (ElementTemplate[], ElementTemplate[], ElementTemplate[], FillMode)	Initializes a new instance of the **Shape** class with the specified outline, decorations and text area.
Shape (ElementTemplate[], ElementTemplate[], ElementTemplate[], FillMode, String)	Initializes a new instance of the **Shape** class with the specified outline, decorations, text area and id.
Shape (ElementTemplate[], ElementTemplate[], ElementTemplate[], FillMode, String, ShapeDecoration[])	Initializes a new instance of the Shape class with the specified outline, decorations, text area and id.
Shape (ElementTemplate[], FillMode)	Initializes a new instance of the **Shape** class with the specified outline.
Shape (ElementTemplate[], FillMode, String)	Initializes a new instance of the **Shape** class with the specified outline and id.
Shape (String, String)	Initializes a new instance of the **Shape** class with the specified outline formula and id.
Shape (String, String, List(ShapeControlPoint), String)	Initializes a new instance of the **Shape** class with the specified outline and decorations formulas, control points and id.
Shape (String, String, String)	Initializes a new instance of the **Shape** class with the specified outline and decorations formulas and id.

See Also

Shape Class
MindFusion.Diagramming Namespace

Shape Constructor (Byte[])

Initializes a new instance of the Shape class with the specified outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    byte[] outline
)
```

Visual Basic Copy Code

```
Public New ( _
    outline() As Byte _
)
```

Parameters

outline
An array containing even number of coordinate values, which define the points of the shape outline.

Remarks

The specified array should contain even number of values. Each odd value represents the X-coordinate and each even value represents the Y-coordinate of the corresponding point in the shape outline. The point coordinates are expressed in numbers in the interval [0..100] indicating the position of this point relative to the node's bounding rectangle.

Exceptions

Exception Type	Condition
ArgumentException	The specified array is a **null** reference (**Nothing** in Visual Basic), does not contain even number of elements or contains less than six arguments.

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace

Shape Constructor (Byte[], String)

Initializes a new instance of the Shape class with the specified outline and id.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    byte[] outline,
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    outline() As Byte, _
    id As String _
)
```

Parameters

outline

An array containing even number of coordinate values, which define the points of the shape outline.

id

The string identifier of the new shape.

Remarks

The specified array should contain even number of values. Each odd value represents the X-coordinate and each even value represents the Y-coordinate of the corresponding point in the shape outline. The point coordinates are expressed in numbers in the interval [0..100] indicating the position of this point relative to the node's bounding rectangle.

Exceptions

Exception Type	Condition
ArgumentException	The specified array is a **null** reference (**Nothing** in Visual Basic), does not contain even number of elements or contains less than six arguments.

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace

Shape Constructor (CreatePathDelegate, CreatePathDelegate, String)

Initializes a new instance of the Shape class with the specified outline and decorations callbacks and id.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    CreatePathDelegate outlineDelegate,
    CreatePathDelegate decorationsDelegate,
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    outlineDelegate As CreatePathDelegate, _
    decorationsDelegate As CreatePathDelegate, _
    id As String _
)
```

Parameters

outlineDelegate
A delegate called to generate **GraphicsPath** objects representing the shape outline.

decorationsDelegate
A delegate called to generate **GraphicsPath** objects representing the shape decorations.

id
The string identifier of the new shape.

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace
ShapeData Class

Shape Constructor (CreatePathDelegate, String)

Initializes a new instance of the Shape class with the specified outline delegate and id.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    CreatePathDelegate outlineDelegate,
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    outlineDelegate As CreatePathDelegate, _
```

```
    id As String _
)
```

Parameters

outlineDelegate

A delegate called to generate **GraphicsPath** objects representing the shape outline.

id

The string identifier of the new shape.

Example

This example creates a triangular shape:

C# Copy Code

```
Shape shape = new Shape(CreatePath, "my shape");

IGraphicsPath CreatePath(Rect shapeData)
{
    var path = GraphicsFactory.CreatePath();
    path.AddLine(10, 0, shapeData.Width, 10);
    path.AddLine(shapeData.Width, 10, shapeData.Width / 2, shapeData.Height);
    path.CloseFigure();
    return path;
}
```

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace
ShapeData Class

Shape Constructor (ElementTemplate[], ElementTemplate[], ElementTemplate[], FillMode)

Initializes a new instance of the Shape class with the specified outline, decorations and text area.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    ElementTemplate[] elements,
    ElementTemplate[] decorations,
    ElementTemplate[] textArea,
    FillMode fillMode
)
```

Visual Basic Copy Code

```
Public New ( _
    elements() As ElementTemplate, _
    decorations() As ElementTemplate, _
    textArea() As ElementTemplate, _
    fillMode As FillMode _
)
```

Parameters

elements

An array of shape template elements that define the outline of a shape. Can contain lines, arcs and Bézier curves.

decorations

An array of shape template elements that define shape decorations. Can contain lines, arcs and Bézier curves. Decorations are visual elements that do not take part in hit testing and clipping.

textArea

An array of shape template elements that define the region of a node in which text is laid out and rendered. Can contain lines, arcs and Bézier curves. Latter two are approximated to polygons by the text-formatting algorithm.

fillMode

Specifies how areas are filled if the outline path intersects itself.

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace

Shape Constructor (ElementTemplate[], ElementTemplate[], ElementTemplate[], FillMode, String)

Initializes a new instance of the Shape class with the specified outline, decorations, text area and id.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    ElementTemplate[] elements,
    ElementTemplate[] decorations,
    ElementTemplate[] textArea,
    FillMode fillMode,
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    elements() As ElementTemplate, _
    decorations() As ElementTemplate, _
    textArea() As ElementTemplate, _
    fillMode As FillMode, _
    id As String _
)
```

Parameters

elements

An array of shape template elements that define the outline of a shape. Can contain lines, arcs and Bézier curves.

decorations

An array of shape template elements that define shape decorations. Can contain lines, arcs and Bézier curves. Decorations are visual elements that do not take part in hit testing and clipping.

textArea

An array of shape template elements that define the region of a node in which text is laid out and rendered. Can contain lines, arcs and Bézier curves. Latter two are approximated to polygons by the text-formatting algorithm.

fillMode

Specifies how areas are filled if the outline path intersects itself.

id

The string identifier of the new shape.

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace

Shape Constructor (ElementTemplate[], ElementTemplate[], ElementTemplate[], FillMode, String, ShapeDecoration[])

Initializes a new instance of the Shape class with the specified outline, decorations, text area and id.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    ElementTemplate[] elements,
    ElementTemplate[] decorations,
    ElementTemplate[] textArea,
    FillMode fillMode,
    string id,
    ShapeDecoration[] closedDecorations
)
```

Visual Basic Copy Code

```
Public New ( _
    elements() As ElementTemplate, _
    decorations() As ElementTemplate, _
    textArea() As ElementTemplate, _
    fillMode As FillMode, _
    id As String, _
    closedDecorations() As ShapeDecoration _
)
```

Parameters

elements
An array of shape template elements that define the outline of a shape. Can contain lines, arcs and Bézier curves.

decorations
An array of shape template elements that define shape decorations. Can contain lines, arcs and Bézier curves. Decorations are visual elements that do not take part in hit testing and clipping.

textArea
An array of shape template elements that define the region of a node in which text is laid out and rendered. Can contain lines, arcs and Bézier curves. Latter two are approximated to polygons by the text-formatting algorithm.

fillMode
Specifies how areas are filled if the outline path intersects itself.

id
The string identifier of the new shape.

closedDecorations
An array of ShapeDecoration objects.

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace

Shape Constructor (ElementTemplate[], FillMode)

Initializes a new instance of the Shape class with the specified outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    ElementTemplate[] elements,
    FillMode fillMode
)
```

Visual Basic Copy Code

```
Public New ( _
    elements() As ElementTemplate, _
    fillMode As FillMode _
)
```

Parameters

elements
An array of shape template elements that define the outline of a shape. Can contain lines, arcs and Bézier curves.

fillMode
Specifies how areas are filled if the outline path intersects itself.

Example

C# Copy Code

```
diagram.DefaultShape = new Shape(
    new ElementTemplate[] {
        new LineTemplate(0, 100, 35, 0),
        new LineTemplate(35, 0, 100, 0),
        new LineTemplate(100, 0, 65, 100),
        new LineTemplate(65, 100, 0, 100) },
    FillMode.Winding);
```

Visual Basic Copy Code

```
diagram.DefaultShape = new Shape( _
    new ElementTemplate() { _
        new LineTemplate(0, 100, 35, 0), _
        new LineTemplate(35, 0, 100, 0), _
        new LineTemplate(100, 0, 65, 100), _
        new LineTemplate(65, 100, 0, 100) }, _
    FillMode.Winding)
```

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace

Shape Constructor (ElementTemplate[], FillMode, String)

Initializes a new instance of the Shape class with the specified outline and id.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    ElementTemplate[] elements,
    FillMode fillMode,
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    elements() As ElementTemplate, _
```

```
    fillMode As FillMode, _
    id As String _
)
```

Parameters

elements
Refers to an array of shape template elements that define the outline of a shape. Can contain lines, arcs and Bézier curves.

fillMode
Specifies how areas are filled if the outline path intersects itself.

id
The string identifier of the new shape.

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace

Shape Constructor (String, String)

Initializes a new instance of the Shape class with the specified outline formula and id.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    string outlineFormula,
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    outlineFormula As String, _
    id As String _
)
```

Parameters

outlineFormula
A string containing the formula that defines the shape.

id
The string identifier of the new shape.

Remarks

Shape formulas are small scripts that call one of the following functions to add graphics to the shape:

Function	Description
MoveTo (x,y)	Moves the current position to the specified point without drawing.
LineTo (x,y)	Draws a line from the current position to the specified point.
BezierTo (x1,y1,x2,y2,x3,y3)	Draws a Bezier curve from the current position to (x3,y3) using (x1,y1) and (x2,y2) as control points.
ArcTo (x,y,largeArc,clockwiseArc,rx,ry)	Draws an arc from the specified point to (x,y) where rx and ry are the ellipse radiuses and the arc flags are boolean values specifying which of the four possible arcs to draw.

Each of them starts drawing from the current position and changes the current position to their point argument after drawing. In addition, the scripts can access the current size of the node through the Width and Height variables.

Example

This sample creates an arrow shape with fixed shaft width and arrowhead length, which will still shrink when the node gets small due to the Min calls.

C# Copy Code

```
string outline = @"
    my = Height / 2;
    ah = Min(10, Height / 4);
    aw = Min(20, Width / 2);
    MoveTo(0, my - ah);
    LineTo(Width - aw, my - ah);
    LineTo(Width - aw, 0);
    LineTo(Width, my);
    LineTo(Width - aw, Height);
    LineTo(Width - aw, my + ah);
    LineTo(0, my + ah);
 ";
Shape shape = new Shape(outline, "my shape");
```

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace
OutlineFormula Property

Shape Constructor (String, String, List(ShapeControlPoint), String)

Initializes a new instance of the Shape class with the specified outline and decorations formulas, control points and id.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    string outlineFormula,
    string decorationsFormula,
    List<ShapeControlPoint> controlPoints,
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    outlineFormula As String, _
    decorationsFormula As String, _
    controlPoints As List(Of ShapeControlPoint), _
    id As String _
)
```

Parameters

outlineFormula
A string containing the formula that defines the shape.

decorationsFormula
A string containing the formula that defines the shape.

controlPoints
A list of ShapeControlPoint objects that let users adjust the shape.

id
The string identifier of the new shape.

Remarks

Shape formulas are small scripts that call one of the following functions to add graphics to the shape:

Function	Description
MoveTo (x,y)	Moves the current position to the specified point without drawing.
LineTo (x,y)	Draws a line from the current position to the specified point.
BezierTo (x1,y1,x2,y2,x3,y3)	Draws a Bezier curve from the current position to (x3,y3) using (x1,y1) and (x2,y2) as control points.
ArcTo (x,y,largeArc,clockwiseArc,rx,ry)	Draws an arc from the specified point to (x,y) where rx and ry are the ellipse radiuses and the arc flags are boolean values specifying which of the four possible arcs to draw.

Each of them starts drawing from the current position and changes the current position to their point argument after drawing. In addition, the scripts can access the current size of the node through the Width and Height variables.

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace
OutlineFormula Property
DecorationsFormula Property
ControlPoints Property

Shape Constructor (String, String, String)

Initializes a new instance of the Shape class with the specified outline and decorations formulas and id.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape (
    string outlineFormula,
    string decorationsFormula,
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    outlineFormula As String, _
    decorationsFormula As String, _
    id As String _
)
```

Parameters

outlineFormula
A string containing the formula that defines the shape outline.

decorationsFormula
A string containing the formula that defines the shape decorations.

id
The string identifier of the new shape.

Remarks

Shape formulas are small scripts that call one of the following functions to add graphics to the shape:

Function	Description
MoveTo (x,y)	Moves the current position to the specified point without drawing.

LineTo (x,y)	Draws a line from the current position to the specified point.
BezierTo (x1,y1,x2,y2,x3,y3)	Draws a Bezier curve from the current position to (x3,y3) using (x1,y1) and (x2,y2) as control points.
ArcTo (x,y,largeArc,clockwiseArc,rx,ry)	Draws an arc from the specified point to (x,y) where rx and ry are the ellipse radiuses and the arc flags are boolean values specifying which of the four possible arcs to draw.

Each of them starts drawing from the current position and changes the current position to their point argument after drawing. In addition, the scripts can access the current size of the node through the Width and Height variables.

See Also

Shape Constructor Overload List
Shape Class
MindFusion.Diagramming Namespace
OutlineFormula Property
DecorationsFormula Property

Shape Methods

Shape.Clone Method

Creates an exact copy of this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As Object
```

Return Value

An instance of the Shape class representing the newly created clone.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.Draw Method

Draws this shape on an IGraphics surface.

Overload List

Name	Description
Shape.Draw (IGraphics, Brush, Pen, Boolean)	Draws this shape on the specified **IGraphics** surface.
Shape.Draw (IGraphics, Brush, Pen, Boolean, Rect, Double)	Draws this shape on the specified **IGraphics** surface.

See Also

Shape.Draw Method (IGraphics, Brush, Pen, Boolean)

Draws this shape on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Draw (
    IGraphics drawingContext,
    Brush brush,
    Pen pen,
    bool shadow
)
```

Visual Basic Copy Code

```
Public Sub Draw( _
    drawingContext As IGraphics, _
    brush As Brush, _
    pen As Pen, _
    shadow As Boolean _
)
```

Parameters

drawingContext
The IGraphics to draw the shape on.

brush
A **Brush** used to fill the shape.

pen
A **Pen** used to draw the shape outline.

shadow
When *shadow* is **true** the *brush* and *pen* parameters have higher priority over shape's brush and pen.

See Also

Draw Method Overload List
Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.Draw Method (IGraphics, Brush, Pen, Boolean, Rect, Double)

Draws this shape on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Draw (
    IGraphics drawingContext,
    Brush brush,
    Pen pen,
    bool shadow,
    Rect rc,
```

```
    double orientation
)
```

Visual Basic Copy Code

```
Public Sub Draw( _
    drawingContext As IGraphics, _
    brush As Brush, _
    pen As Pen, _
    shadow As Boolean, _
    rc As Rect, _
    orientation As Double _
)
```

Parameters

drawingContext
The IGraphics to draw the shape on.

brush
A **Brush** used to fill the shape.

pen
A **Pen** used to draw the shape outline.

shadow
When *shadow* is **true** the *brush* and *pen* parameters have higher priority over shape's brush and pen.

rc
The bounding rectangle.

orientation
Orientation of the shape.

See Also

Draw Method Overload List
Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.Equals Method

Checks whether this shape template is equivalent to the specified shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public new bool Equals (
    Shape shape
)
```

Visual Basic Copy Code

```
Public Shadows Function Equals( _
    shape As Shape _
) As Boolean
```

Parameters

shape
The shape to check for equality with the current shape.

Return Value

true if the current shape is equivalent to the specified shape; otherwise, **false**.

Remarks

Two shapes are considered equivalent if they have the same fill type, the same number and type of segments and if the points of those segments are close enough to each other.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.FromId Method

Returns a reference to the shape with the specified string identifier.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape FromId (
    string id
)
```

Visual Basic Copy Code

```
Public Shared Function FromId( _
    id As String _
) As Shape
```

Parameters

id
The identifier of the shape of interest.

Return Value

A reference to the Shape with the specified identifier or a **null** reference (**Nothing** in Visual Basic) if there is no such shape.

Remarks

The *id* argument should match an identifier passed to a constructor of the existing shape templates.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.FromSvg Method

Creates a Shape from the outline of an SVG drawing.

Overload List

Name	Description
Shape.FromSvg (SvgContent)	Creates a **Shape** from the outline of an SVG drawing.
Shape.FromSvg (SvgContent, Boolean)	Creates a **Shape** from the outline of an SVG drawing.
Shape.FromSvg (SvgContent, Boolean, Boolean)	Creates a Shape from the outline of an SVG drawing.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.FromSvg Method (SvgContent)

Creates a Shape from the outline of an SVG drawing.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape FromSvg (
    SvgContent svgContent
)
```

Visual Basic Copy Code

```
Public Shared Function FromSvg( _
    svgContent As SvgContent _
) As Shape
```

Parameters

svgContent
An SvgContent object representing the SVG drawing.

Return Value

A Shape generated from the specified SVG drawing.

Remarks

SVG drawings do not contain intrinsic semantic information about which elements of a drawing should be considered its outline. However the Shape class is primarily concerned with the outlines of nodes, for drawing, hit-testing and link alignment purposes. If you need the filled interior of nodes and the positions of links connected to them to match the SVG content, call this method to generate a shape outline from SVG, and assign the result to the nodes' Shape property. The shape outline is generated using a convex-hull algorithm.

See Also

FromSvg Method Overload List
Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.FromSvg Method (SvgContent, Boolean)

Creates a Shape from the outline of an SVG drawing.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape FromSvg (
    SvgContent svgContent,
    bool stretch
)
```

Visual Basic Copy Code

```
Public Shared Function FromSvg( _
    svgContent As SvgContent, _
```

```
    stretch As Boolean _
) As Shape
```

Parameters

svgContent
An SvgContent object representing the SVG drawing.

stretch
Sets whether the Shape outline generated from SVG should be stretched to fill the entire shape definition range [0-100, 0-100], or should be scaled uniformly to preserve the original aspect ratio of the drawing. In effect, this specifies whether the shape will match the SVG drawing when ContentAlign is set to Stretch or **Fit**.

Return Value

A **Shape** generated from the specified SVG drawing.

Remarks

SVG drawings do not contain intrinsic semantic information about which elements of a drawing should be considered its outline. However the Shape class is primarily concerned with the outlines of nodes, for drawing, hit-testing and link alignment purposes. If you need the filled interior of nodes and the positions of links connected to them to match the SVG content, call this method to generate a shape outline from SVG, and assign the result to the nodes' Shape property. The shape outline is generated using a convex-hull algorithm.

See Also

FromSvg Method Overload List
Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.FromSvg Method (SvgContent, Boolean, Boolean)

Creates a Shape from the outline of an SVG drawing.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape FromSvg (
    SvgContent svgContent,
    bool stretch,
    bool concave
)
```

Visual Basic Copy Code

```
Public Shared Function FromSvg( _
    svgContent As SvgContent, _
    stretch As Boolean, _
    concave As Boolean _
) As Shape
```

Parameters

svgContent
An SvgContent object representing the SVG drawing.

stretch
Specifies whether the Shape outline should match the SVG drawing when ContentAlign is set to Stretch.

concave
Specifies whether the Shape outline should be defined by a concave hull.

Return Value

A Shape generated from the specified SVG drawing.

See Also

FromSvg Method Overload List
Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.GetHitPath Method

Returns a **GraphicsPath** object corresponding to the outline of the specified shape when rotated at the specified angle. That **GraphicsPath** can be used for hit-testing and intersection checks.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IGraphicsPath GetHitPath (
    Rect nodeBounds,
    double rotationAngle,
    double orientation,
    bool flipX,
    bool flipY,
    Dictionary<string, ShapeFormulaParam> formulaParams
)
```

Visual Basic Copy Code

```
Public Function GetHitPath( _
    nodeBounds As Rect, _
    rotationAngle As Double, _
    orientation As Double, _
    flipX As Boolean, _
    flipY As Boolean, _
    formulaParams As Dictionary(Of String, ShapeFormulaParam) _
) As IGraphicsPath
```

Parameters

nodeBounds
The bounding rectangle of the ShapeNode.

rotationAngle
The node's rotation angle.

orientation
The shape's orientation angle.

flipX
true to flip the path horizontally, or **false** otherwise.

flipY
true to flip the path vertically, or **false** otherwise.

formulaParams
A list of ShapeFormulaParam objects specifying the current positions of shape control points.

Return Value

A **GraphicsPath** representing the shape's outline.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.GetRenderingPath Method

Returns a **GraphicsPath** object corresponding to the outline of the shape when rotated at the specified angle. That **GraphicsPath** can be used when drawing the shape's outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IGraphicsPath GetRenderingPath (
    Rect nodeBounds,
    double rotationAngle,
    double orientation,
    bool flipX,
    bool flipY,
    Dictionary<string, ShapeFormulaParam> formulaParams
)
```

Visual Basic Copy Code

```
Public Function GetRenderingPath( _
    nodeBounds As Rect, _
    rotationAngle As Double, _
    orientation As Double, _
    flipX As Boolean, _
    flipY As Boolean, _
    formulaParams As Dictionary(Of String, ShapeFormulaParam) _
) As IGraphicsPath
```

Parameters

nodeBounds
The bounding rectangle of the ShapeNode.

rotationAngle
The node's rotation angle.

orientation
The shape's orientation angle.

flipX
true to flip the path horizontally, or **false** otherwise.

flipY
true to flip the path vertically, or **false** otherwise.

formulaParams
A list of ShapeFormulaParam objects specifying the current positions of shape control points.

Return Value

A **GraphicsPath** representing the shape's outline.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.LoadFromXml Method

Loads the shape content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic

```
Public Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

The **XmlElement** containing the shape's data.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace
SaveToXml Method

Shape.SaveTo Method

Overload List

Name	Description

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.SaveToXml Method

Saves the shape content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic

```
Public Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

The **XmlElement** the shape's data should be stored into.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace
LoadFromXml Method

Shape.ToString Method

Returns the identifier of this shape template.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override string ToString ()
```

Visual Basic Copy Code

```
Public Overrides Function ToString () As String
```

Return Value

A string value representing the shape identifier.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape Properties

Shape.AnchorPattern Property

Gets or sets the default anchor pattern for nodes that use this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern AnchorPattern { get; set; }
```

Visual Basic Copy Code

```
Public Property AnchorPattern As AnchorPattern
```

Property Value

An AnchorPattern instance. The default value is **null** (**Nothing** in Visual Basic).

Remarks

If the **AnchorPattern** of a ShapeNode is set, it overrides the one associated with the Shape definition.

See Also

Shape.ControlPoints Property

A list of ShapeControlPoint objects that let users adjust a node's shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<ShapeControlPoint> ControlPoints { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ControlPoints As List(Of ShapeControlPoint)
```

Property Value

A List<ShapeControlPoint> instance.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.DecorationHitTesting Property

Gets or sets a value indicating whether decoration lines should be considered when hit-testing.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool DecorationHitTesting { get; set; }
```

Visual Basic Copy Code

```
Public Property DecorationHitTesting As Boolean
```

Property Value

true if decoration lines should be considered for hit-testing; otherwise, **false**. The default is **false**.

Remarks

If this property is enabled, users are able to select shape nodes of such shape by clicking near a decoration line or curve, even if it lies outside the shape outlines.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.Decorations Property

Gets or sets a list with all decorations of this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ElementTemplate[] Decorations { get; set; }
```

Visual Basic Copy Code

```
Public Property Decorations As ElementTemplate()
```

Property Value

An array of ElementTemplate objects specifying the decorations of this shape.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace
Outline Property
DecorationsFormula Property

Shape.DecorationsFormula Property

Gets or sets a formula describing the shape decorations depending on the current width and height of the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string DecorationsFormula { get; set; }
```

Visual Basic Copy Code

```
Public Property DecorationsFormula As String
```

Property Value

A string containing the script that defines the shape decorations.

Remarks

Shape formulas are small scripts that call one of the following functions to add graphics to the shape:

Function	Description
MoveTo (x,y)	Moves the current position to the specified point without drawing.
LineTo (x,y)	Draws a line from the current position to the specified point.
BezierTo (x1,y1,x2,y2,x3,y3)	Draws a Bezier curve from the current position to (x3,y3) using (x1,y1) and (x2,y2) as control points.
ArcTo (x,y,largeArc,clockwiseArc,rx,ry)	Draws an arc from the specified point to (x,y) where rx and ry are the ellipse radiuses and the arc flags are boolean values specifying which of the four possible arcs to draw.

Each of them starts drawing from the current position and changes the current position to their point argument after drawing. In addition, the scripts can access the current size of the node through the Width and Height variables.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace
Decorations Property
OutlineFormula Property

Shape.DisplayName Property

Specifies the name that should be displayed for this shape in user interface objects, for example the **ShapeListBox** or **ShapeToolBar**.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string DisplayName { get; set; }
```

Visual Basic Copy Code

```
Public Property DisplayName As String
```

Property Value

The **DisplayName** of all predefined shapes is set to the value of their Id property.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.EnableOutline Property

Gets or sets a value indicating whether to enable the shape's outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableOutline { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableOutline As Boolean
```

Property Value

Remarks

When

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.FillMode Property

Gets or sets a value indicating how areas are filled if the outline path of the shape intersects itself.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public FillMode FillMode { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property FillMode As FillMode
```

Property Value

A member of the FillMode enumeration specifying the fill type of the shape.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.Id Property

Gets or sets the shape identifier.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

A string id that has been passed as an argument of the Shape constructor.

Remarks

Once assigned through the constructor, the shape identifier should not be changed, that is, do not use the *set* accessor of this property. To get the shape with a specified id, call the static FromId method.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.Image Property

Gets or sets the image displayed in all instances of this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage Image { get; set; }
```

Visual Basic Copy Code

```
Public Property Image As IImage
```

Property Value

A **Microsoft.Maui.Graphics.IImage** instance. The default is **null** (**Nothing** in Visual Basic).

Remarks

Assigning an image object to this property sets the ImageLocation to **null** (**Nothing** in Visual Basic). The image is displayed in the rectangle specified in ImageRectangle. If the Image property of a shape node is set, it overrides the image specified in the shape definition.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.ImageLocation Property

Gets or sets the path to the file that contains the image displayed in this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ImageLocation { get; set; }
```

Visual Basic Copy Code

```
Public Property ImageLocation As String
```

Property Value

A string containing the file path. The default is **null** (**Nothing** in Visual Basic).

Remarks

Assigning a valid file path to this property will automatically load the image. When adding a shape to a ShapeLibrary, if the Image property is set and the image location is not specified, the image bitmap contents is stored in the library file. If the image location is specified, only the path to the image file is saved in the shape library.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.ImageRectangle Property

Gets or sets the position of the images displayed in instances of this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect ImageRectangle { get; set; }
```

Visual Basic Copy Code

```
Public Property ImageRectangle As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** specifying the location of an image relatively to the shape node that displays the image.

Remarks

The coordinates of the image location are specified as percentage of the rectangular bounds of shape nodes. The default value (0, 0, 100, 100) makes the image displayed in the whole area of that node.

Example

The following code defines a shape, which displays an image in the lower-right quarter of shape nodes and lays out text in the other three quarters.

C# Copy Code

```
Shape myShape = new Shape(

    // Outlines: reuse the rectangle definition
    Shape.FromId("Rectangle").Outline,

    // Decorations: draw frame lines around the image
    new ElementTemplate[] {
        new LineTemplate(100, 50, 50, 50),
        new LineTemplate(50, 50, 50, 100),
    },

    // Text area: the area not covered by the image rectangle
    new ElementTemplate[] {
        new LineTemplate(0, 0, 100, 0),
        new LineTemplate(100, 0, 100, 50),
        new LineTemplate(100, 50, 50, 50),
        new LineTemplate(50, 50, 50, 100),
        new LineTemplate(50, 100, 0, 100),
        new LineTemplate(0, 100, 0, 0)
    },

    // The fill type
    FillMode.Winding,

    // The id
    "MyShape");

// Image area: the lower right corner of boxes
myShape.ImageRectangle = new Rect(50, 50, 50, 50);

// Use the shape defined above
diagram.DefaultShape = myShape;

// Use polygonal text layout
diagram.EnableStyledText = true;
diagram.PolygonalTextLayout = true;
```

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace
Image Property
Image Property

Shape.IsArrowhead Property

Specifies if this shape is used as arrowhead.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsArrowhead { get; set; }
```

Visual Basic Copy Code

```
Public Property IsArrowhead As Boolean
```

Property Value

true if this shape is designated as arrowhead; otherwise **false**.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.LinkSegmentInset Property

Specifies what length of link's line segment is drawn inside the arrowhead shape, expressed as percentage of arrowhead size.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double LinkSegmentInset { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkSegmentInset As Double
```

Property Value

A **double** value specifying percents of arrowhead size. The default value is 100.

Remarks

The inset value matters when using transparent or semi-transparent arrowhead brush, or when the link's Pen is thicker than one pixel and its cap might extend beyond the arrowhead outline. **LinkSegmentInset** is set to appropriate values for all shapes from the ArrowHeads class. You might want to set it when using custom arrowhead shapes for which the link's end segment should not reach all the way to the arrowhead tip.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.Outline Property

Gets or sets the outline of this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ElementTemplate[] Outline { get; set; }
```

Visual Basic Copy Code

```
Public Property Outline As ElementTemplate()
```

Property Value

An array of ElementTemplate objects specifying the outline of the shape.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace
Decorations Property
OutlineFormula Property

Shape.OutlineFormula Property

Gets or sets a formula describing the shape outline depending on the current width and height of the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string OutlineFormula { get; set; }
```

Visual Basic Copy Code

```
Public Property OutlineFormula As String
```

Property Value

A string containing the script that defines the shape outline.

Remarks

Shape formulas are small scripts that call one of the following functions to add graphics to the shape:

Function	Description
MoveTo (x,y)	Moves the current position to the specified point without drawing.
LineTo (x,y)	Draws a line from the current position to the specified point.
BezierTo (x1,y1,x2,y2,x3,y3)	Draws a Bezier curve from the current position to (x3,y3) using (x1,y1) and (x2,y2) as control points.
ArcTo (x,y,largeArc,clockwiseArc,rx,ry)	Draws an arc from the specified point to (x,y) where rx and ry are the ellipse radiuses and the arc flags are boolean values specifying which of the four possible arcs to draw.

Each of them starts drawing from the current position and changes the current position to their point argument after drawing. In addition, the scripts can access the current size of the node through the Width and Height variables.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace
Outline Property
DecorationsFormula Property

Shape.ShapeDecorations Property

Gets or sets the ShapeDecoration objects associated with this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeDecoration[] ShapeDecorations { get; set; }
```

Visual Basic Copy Code

```
Public Property ShapeDecorations As ShapeDecoration()
```

Property Value

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

Shape.Shapes Property

Gets a read-only collection of all named shapes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static ShapeCollection Shapes { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Shapes As ShapeCollection
```

Property Value

An instance of the ShapeCollection class.

Remarks

Initially the collection contains only the predefined shapes. Any new shape you create is added automatically to **Shapes** if an identifier has been specified via the Shape constructor.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace
FromId Method

Shape.TextArea Property

Gets or sets the outline of the text area of this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ElementTemplate[] TextArea { get; set; }
```

Visual Basic Copy Code

```
Public Property TextArea As ElementTemplate()
```

Property Value

An array of ElementTemplate objects specifying the text area of this shape.

See Also

Shape Members
Shape Class
MindFusion.Diagramming Namespace

ShapeAdjustmentHandle Class

Represents an adjustment handle used to modify the Shape of a ShapeNode.

Syntax

C# Copy Code

```
public class ShapeAdjustmentHandle : AdjustmentHandle
```

Visual Basic Copy Code

```
Public Class ShapeAdjustmentHandle
    Inherits AdjustmentHandle
```

Remarks

Adjustment handles of this type are used to modify the ControlPoints of a formula shape.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.AdjustmentHandle
    MindFusion.Diagramming.ShapeAdjustmentHandle

See Also

ShapeAdjustmentHandle Members
MindFusion.Diagramming Namespace

ShapeAdjustmentHandle Members

Represents an adjustment handle used to modify the Shape of a ShapeNode.

The following tables list the members exposed by the ShapeAdjustmentHandle type.

Public Constructors

	Name	Description
	ShapeAdjustmentHandle	Initializes a new instance of the **ShapeAdjustmentHandle** class.

Public Properties

	Name	Description

	ControlPoint	Gets or sets the ShapeControlPoint that will be modified by this adjustment handle.
	FormulaParam	Gets or sets the ShapeFormulaParam whose values will be updated by this handle.

Public Methods

	Name	Description
	IsMoveHandle	Returns a value indicating whether this handle is used to move items. (Inherited from AdjustmentHandle.)
	IsResizeHandle	Gets whether this adjustment handle resizes the item. (Inherited from AdjustmentHandle.)
	IsRotationHandle	Gets whether this adjustment handle changes item's rotation angle. (Inherited from AdjustmentHandle.)

See Also

ShapeAdjustmentHandle Class
MindFusion.Diagramming Namespace

ShapeAdjustmentHandle Constructor

Initializes a new instance of the ShapeAdjustmentHandle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeAdjustmentHandle (
    ShapeControlPoint controlPoint,
    ShapeFormulaParam formulaParam
)
```

Visual Basic Copy Code

```
Public New ( _
    controlPoint As ShapeControlPoint, _
    formulaParam As ShapeFormulaParam _
)
```

Parameters

controlPoint
A ShapeControlPoint representing the shape's control point that will be modified by this adjustment handle.

formulaParam
A ShapeFormulaParam whose values will be updated by this handle.

See Also

ShapeAdjustmentHandle Class
MindFusion.Diagramming Namespace

ShapeAdjustmentHandle Properties

ShapeAdjustmentHandle.ControlPoint Property

Gets or sets the ShapeControlPoint that will be modified by this adjustment handle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeControlPoint ControlPoint { get; set; }
```

Visual Basic Copy Code

```
Public Property ControlPoint As ShapeControlPoint
```

Property Value

A ShapeControlPoint instance.

See Also

ShapeAdjustmentHandle Members
ShapeAdjustmentHandle Class
MindFusion.Diagramming Namespace

ShapeAdjustmentHandle.FormulaParam Property

Gets or sets the ShapeFormulaParam whose values will be updated by this handle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeFormulaParam FormulaParam { get; set; }
```

Visual Basic Copy Code

```
Public Property FormulaParam As ShapeFormulaParam
```

Property Value

A ShapeFormulaParam instance.

See Also

ShapeAdjustmentHandle Members
ShapeAdjustmentHandle Class
MindFusion.Diagramming Namespace

ShapeCollection Class

Represents a read-only collection of shapes.

Syntax

C# Copy Code

```
public class ShapeCollection : ICollection, IEnumerable
```

Visual Basic Copy Code

```
Public Class ShapeCollection
    Implements ICollection, IEnumerable
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ShapeCollection

See Also

ShapeCollection Members
MindFusion.Diagramming Namespace

ShapeCollection Members

Represents a read-only collection of shapes.

The following tables list the members exposed by the ShapeCollection type.

Public Properties

	Name	Description
	Count	Gets the number of elements contained in the collection.
	IsSynchronized	Gets a value indicating whether the access to this collection is synchronized.
	Item	Gets or sets the diagram item at the specified position.
	SyncRoot	Gets an object that can be used to synchronize access to this collection.

Public Methods

	Name	Description
	Contains	Checks whether the specified shape is contained in the collection.
	CopyTo	Copies a range of shapes from this collection to the specified one-dimensional array starting at the specified index of the target array.
	GetAt	Gets the shape at the specified position.
	GetEnumerator	Returns an enumerator for this collection.

See Also

ShapeCollection Class
MindFusion.Diagramming Namespace

ShapeCollection Methods

ShapeCollection.Contains Method

Checks whether the specified shape is contained in the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Contains (
    Shape shape
)
```

Visual Basic Copy Code

```
Public Function Contains( _
    shape As Shape _
) As Boolean
```

Parameters

shape

The Shape object to search for in the collection.

Return Value

true if the specified item was found in the collection; otherwise, **false**.

See Also

ShapeCollection Members
ShapeCollection Class
MindFusion.Diagramming Namespace

ShapeCollection.CopyTo Method

Copies a range of shapes from this collection to the specified one-dimensional array starting at the specified index of the target array.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void CopyTo (
    Array array,
    int index
)
```

Visual Basic Copy Code

```
Public Sub CopyTo( _
    array As Array, _
    index As Integer _
)
```

Parameters

array

The one-dimensional **Array** that is the destination of the elements copied from the current collection.

index

A 32-bit integer that represents the index in array at which copying begins.

See Also

ShapeCollection Members
ShapeCollection Class
MindFusion.Diagramming Namespace

ShapeCollection.GetAt Method

Gets the shape at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape GetAt (
    int index
)
```

Visual Basic Copy Code

```
Public Function GetAt( _
    index As Integer _
) As Shape
```

Parameters

index
The zero-based index of the shape to retrieve.

Return Value

An instance of the Shape class that represents the shape at the specified position.

See Also

ShapeCollection Members
ShapeCollection Class
MindFusion.Diagramming Namespace

ShapeCollection.GetEnumerator Method

Returns an enumerator for this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IEnumerator GetEnumerator ()
```

Visual Basic Copy Code

```
Public Function GetEnumerator () As IEnumerator
```

Return Value

An .NET **IEnumerator** for the collection.

See Also

ShapeCollection Members
ShapeCollection Class
MindFusion.Diagramming Namespace

ShapeCollection Properties

ShapeCollection.Count Property

Gets the number of elements contained in the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Count { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Count As Integer
```

Property Value

An integer value specifying the size of the collection.

See Also

ShapeCollection Members
ShapeCollection Class
MindFusion.Diagramming Namespace

ShapeCollection.IsSynchronized Property

Gets a value indicating whether the access to this collection is synchronized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsSynchronized { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property IsSynchronized As Boolean
```

Property Value

true if access to the collection is synchronized (thread safe); otherwise, **false**. The default is **false**.

See Also

ShapeCollection Members
ShapeCollection Class
MindFusion.Diagramming Namespace

ShapeCollection.Item Property

Gets or sets the diagram item at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape Item [
    int index
] { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Item( _
    index As Integer _
) As Shape
```

Parameters

index
The zero-based index of the shape to retrieve.

Property Value

An instance of the Shape class that represents the shape at the specified position.

See Also

ShapeCollection Members
ShapeCollection Class
MindFusion.Diagramming Namespace

ShapeCollection.SyncRoot Property

Gets an object that can be used to synchronize access to this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object SyncRoot { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property SyncRoot As Object
```

Property Value

An object that can be used to synchronize access to the collection.

See Also

ShapeCollection Members
ShapeCollection Class
MindFusion.Diagramming Namespace

ShapeControlPoint Class

Represents a control point that can be used to adjust the shape of a ShapeNode. Control point positions are passed as parameters to shape formulas, whose code can alter the coordinates of the shape's graphic primitives based on the parameter values.

Syntax

C# Copy Code

```
public class ShapeControlPoint
```

Visual Basic Copy Code

```
Public Class ShapeControlPoint
```

Remarks

The component raises the ShapeModified event when users drag a shape control point.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.ShapeControlPoint

See Also

ShapeControlPoint Members
MindFusion.Diagramming Namespace

ShapeControlPoint Members

Represents a control point that can be used to adjust the shape of a ShapeNode. Control point positions are passed as parameters to shape formulas, whose code can alter the coordinates of the shape's graphic primitives based on the parameter values.

The following tables list the members exposed by the ShapeControlPoint type.

Public Constructors

	Name	Description
	ShapeControlPoint	Initializes a new instance of the **ShapeControlPoint** class.

Public Properties

	Name	Description
	MaxXRelativeToWidth	Gets or sets a value indicating whether the maximum horizontal position is specified relatively to the node's right side.
	MaxYRelativeToHeight	Gets or sets a value indicating whether the maximum vertical position is specified relatively to the node's bottom side.
	ParamName	Gets or sets the name of the formula parameter corresponding to this control point.

Public Methods

	Name	Description
	Clone	Creates a copy of this control point.

See Also

ShapeControlPoint Class
MindFusion.Diagramming Namespace

ShapeControlPoint Constructor

Initializes a new instance of the ShapeControlPoint class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeControlPoint (
    string paramName,
    double defaultX,
    double minX,
    double maxX,
    UnitType xUnit,
    double defaultY,
    double minY,
    double maxY,
```

```
    UnitType yUnit
)
```

Visual Basic Copy Code

```
Public New ( _
    paramName As String, _
    defaultX As Double, _
    minX As Double, _
    maxX As Double, _
    xUnit As UnitType, _
    defaultY As Double, _
    minY As Double, _
    maxY As Double, _
    yUnit As UnitType _
)
```

Parameters

paramName
Specifies the name of the shape formula parameter corresponding to this control point.

defaultX
The initial horizontal position of this control point.

minX
The minimum horizontal position that is allowed for this control point.

maxX
The maximum horizontal position that is allowed for this control point.

xUnit
The unit used to specify the horizontal position of this control point.

defaultY
The initial vertical position of this control point.

minY
The minimum vertical position that is allowed for this control point.

maxY
The maximum vertical position that is allowed for this control point.

yUnit
The unit used to specify the vertical position of this control point.

See Also

ShapeControlPoint Class
MindFusion.Diagramming Namespace

ShapeControlPoint Methods

ShapeControlPoint.Clone Method

Creates a copy of this control point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeControlPoint Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As ShapeControlPoint
```

Return Value

A new ShapeControlPoint that is a copy of the current instance.

See Also

ShapeControlPoint Members
ShapeControlPoint Class
MindFusion.Diagramming Namespace

ShapeControlPoint Properties

ShapeControlPoint.MaxXRelativeToWidth Property

Gets or sets a value indicating whether the maximum horizontal position is specified relatively to the node's right side.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool MaxXRelativeToWidth { get; set; }
```

Visual Basic Copy Code

```
Public Property MaxXRelativeToWidth As Boolean
```

Property Value

true if the *maxX* value is relative to the node's right side, **false** if it's relative to the left side.

See Also

ShapeControlPoint Members
ShapeControlPoint Class
MindFusion.Diagramming Namespace

ShapeControlPoint.MaxYRelativeToHeight Property

Gets or sets a value indicating whether the maximum vertical position is specified relatively to the node's bottom side.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool MaxYRelativeToHeight { get; set; }
```

Visual Basic Copy Code

```
Public Property MaxYRelativeToHeight As Boolean
```

Property Value

true if the *maxY* value is relative to the node's bottom side, **false** if it's relative to the top side.

See Also

ShapeControlPoint Members
ShapeControlPoint Class
MindFusion.Diagramming Namespace

ShapeControlPoint.ParamName Property

Gets or sets the name of the formula parameter corresponding to this control point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ParamName { get; set; }
```

Visual Basic Copy Code

```
Public Property ParamName As String
```

Property Value

A string specifying the name of the shape formula parameter.

See Also

ShapeControlPoint Members
ShapeControlPoint Class
MindFusion.Diagramming Namespace

ShapeConverter Class

Converts from Shape to other data types.

Syntax

C# Copy Code

```
public class ShapeConverter : TypeConverter
```

Visual Basic Copy Code

```
Public Class ShapeConverter
    Inherits TypeConverter
```

Inheritance Hierarchy

System.Object
  System.ComponentModel.TypeConverter
    MindFusion.Diagramming.ShapeConverter

See Also

ShapeConverter Members
MindFusion.Diagramming Namespace

ShapeConverter Members

Converts from Shape to other data types.

The following tables list the members exposed by the ShapeConverter type.

Public Constructors

	Name	Description
	ShapeConverter	Initializes a new instance of the **ShapeConverter** class.

Public Methods

	Name	Description
	CanConvertFrom	**TypeConverter.CanConvertFrom** override.
	CanConvertTo	**TypeConverter.CanConvertTo** override.
	ConvertFrom	**TypeConverter.ConvertFrom** override.
	ConvertTo	**TypeConverter.ConvertTo** override.
	GetStandardValues	**TypeConverter.GetStandardValues** override.
	GetStandardValuesExclusive	**TypeConverter.GetStandardValuesExclusive** override.
	GetStandardValuesSupported	**TypeConverter.GetStandardValuesSupported** override.

See Also

ShapeConverter Class
MindFusion.Diagramming Namespace

ShapeConverter Constructor

Initializes a new instance of the ShapeConverter class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeConverter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ShapeConverter Class
MindFusion.Diagramming Namespace

ShapeConverter Methods

ShapeConverter.CanConvertFrom Method

TypeConverter.CanConvertFrom override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool CanConvertFrom (
    ITypeDescriptorContext context,
    Type sourceType
)
```

Visual Basic

```
Public Overrides Function CanConvertFrom( _
    context As ITypeDescriptorContext, _
    sourceType As Type _
) As Boolean
```

Parameters

context

An **ITypeDescriptorContext** that provides a format context.

sourceType

A **Type** that represents the type you want to convert from.

Return Value

true if this converter can perform the conversion; otherwise, **false**.

See Also

ShapeConverter Members
ShapeConverter Class
MindFusion.Diagramming Namespace

ShapeConverter.CanConvertTo Method

TypeConverter.CanConvertTo override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override bool CanConvertTo (
    ITypeDescriptorContext context,
    Type destinationType
)
```

Visual Basic

```
Public Overrides Function CanConvertTo( _
    context As ITypeDescriptorContext, _
    destinationType As Type _
) As Boolean
```

Parameters

context

An **ITypeDescriptorContext** that provides a format context.

destinationType

A **Type** that represents the type you want to convert to.

Return Value

true if this converter can perform the conversion; otherwise, **false**.

See Also

ShapeConverter Members
ShapeConverter Class
MindFusion.Diagramming Namespace

ShapeConverter.ConvertFrom Method

TypeConverter.ConvertFrom override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Object ConvertFrom (
    ITypeDescriptorContext context,
    CultureInfo culture,
    Object value
)
```

Visual Basic Copy Code

```
Public Overrides Function ConvertFrom( _
    context As ITypeDescriptorContext, _
    culture As CultureInfo, _
    value As Object _
) As Object
```

Parameters

context
An **ITypeDescriptorContext** that provides a format context.

culture
The **CultureInfo** to use as the current culture.

value
The **Object** to convert.

Return Value

An **Object** that represents the converted value.

See Also

ShapeConverter Members
ShapeConverter Class
MindFusion.Diagramming Namespace

ShapeConverter.ConvertTo Method

TypeConverter.ConvertTo override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Object ConvertTo (
    ITypeDescriptorContext context,
    CultureInfo culture,
    Object value,
    Type destinationType
)
```

Visual Basic Copy Code

```
Public Overrides Function ConvertTo( _
    context As ITypeDescriptorContext, _
    culture As CultureInfo, _
    value As Object, _
    destinationType As Type _
) As Object
```

Parameters

context
An **ITypeDescriptorContext** that provides a format context.

culture

A **CultureInfo**. If **null** is passed, the current culture is assumed.

value
The **Object** to convert.

destinationType
An **Object** that represents the converted value.

Return Value

An **Object** that represents the converted value.

See Also

ShapeConverter Members
ShapeConverter Class
MindFusion.Diagramming Namespace

ShapeConverter.GetStandardValues Method

TypeConverter.GetStandardValues override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override TypeConverter.StandardValuesCollection GetStandardValues (
    ITypeDescriptorContext context
)
```

Visual Basic Copy Code

```
Public Overrides Function GetStandardValues( _
    context As ITypeDescriptorContext _
) As TypeConverter.StandardValuesCollection
```

Parameters

context
An **System.ComponentModel.ITypeDescriptorContext** that provides a format context that can be used to extract additional information about the environment from which this converter is invoked. This parameter or properties of this parameter can be **null**.

Return Value

A **System.ComponentModel.TypeConverter.StandardValuesCollection** that holds a standard set of valid values.

See Also

ShapeConverter Members
ShapeConverter Class
MindFusion.Diagramming Namespace

ShapeConverter.GetStandardValuesExclusive Method

TypeConverter.GetStandardValuesExclusive override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool GetStandardValuesExclusive (
    ITypeDescriptorContext context
)
```

Visual Basic

```
Public Overrides Function GetStandardValuesExclusive( _
    context As ITypeDescriptorContext _
) As Boolean
```

Parameters

context
An **System.ComponentModel.ITypeDescriptorContext** that provides a format context.

Return Value

Always returns **true**.

See Also

ShapeConverter Members
ShapeConverter Class
MindFusion.Diagramming Namespace

ShapeConverter.GetStandardValuesSupported Method

TypeConverter.GetStandardValuesSupported override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override bool GetStandardValuesSupported (
    ITypeDescriptorContext context
)
```

Visual Basic

```
Public Overrides Function GetStandardValuesSupported( _
    context As ITypeDescriptorContext _
) As Boolean
```

Parameters

context
An **System.ComponentModel.ITypeDescriptorContext** that provides a format context.

Return Value

Always returns **true**.

See Also

ShapeConverter Members
ShapeConverter Class
MindFusion.Diagramming Namespace

ShapeData Class

Contains the cached coordinates of the key points of a shape when the shape is applied to a specific ShapeNode. This class also stores additional data passed to custom shape-generation callbacks and to shape formula scripts.

Syntax

C#

```
public class ShapeData : ICloneable
```

Visual Basic

```
Public Class ShapeData
    Implements ICloneable
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ShapeData

See Also

ShapeData Members
MindFusion.Diagramming Namespace

ShapeData Members

Contains the cached coordinates of the key points of a shape when the shape is applied to a specific ShapeNode. This class also stores additional data passed to custom shape-generation callbacks and to shape formula scripts.

The following tables list the members exposed by the ShapeData type.

Public Constructors

	Name	Description
	ShapeData	Initializes a new instance of the **ShapeData** class.

Public Properties

	Name	Description
	Height	Gets the height of the node that displays this shape.
	Width	Gets the width of the node that displays this shape.

Public Methods

	Name	Description
	Clone	Creates an exact copy of this **ShapeData** object.

See Also

ShapeData Class
MindFusion.Diagramming Namespace

ShapeData Constructor

Initializes a new instance of the ShapeData class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeData (
    int numElements
)
```

Visual Basic Copy Code

```
Public New ( _
    numElements As Integer _
)
```

Parameters

numElements
Specifies the length of the coordinate cache array.

See Also

ShapeData Class
MindFusion.Diagramming Namespace

ShapeData Methods

ShapeData.Clone Method

Creates an exact copy of this ShapeData object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As Object
```

Return Value

A ShapeData instance.

See Also

ShapeData Members
ShapeData Class
MindFusion.Diagramming Namespace

ShapeData Properties

ShapeData.Height Property

Gets the height of the node that displays this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Height { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Height As Double
```

Property Value

A **double** value specifying the node's height.

See Also

ShapeData Members
ShapeData Class
MindFusion.Diagramming Namespace

ShapeData.Width Property

Gets the width of the node that displays this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Width { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Width As Double
```

Property Value

A **double** value specifying the node's width.

See Also

ShapeData Members
ShapeData Class
MindFusion.Diagramming Namespace

ShapeDecoration Class

Defines decoration elements that are displayed as filled shapes.

Syntax

C# Copy Code

```
public class ShapeDecoration
```

Visual Basic Copy Code

```
Public Class ShapeDecoration
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ShapeDecoration

See Also

ShapeDecoration Members
MindFusion.Diagramming Namespace

ShapeDecoration Members

Defines decoration elements that are displayed as filled shapes.

The following tables list the members exposed by the ShapeDecoration type.

Public Constructors

	Name	Description
	ShapeDecoration	Initializes a new instance of the ShapeDecoration class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the decoration shape interior.
	Elements	Gets the elements that define the decoration's shape.
	FillMode	Gets or sets how areas are filled if the outline path intersects itself.
	Pen	Gets or sets the brush used to draw the decoration shape outline.

Public Methods

	Name	Description
	Clone	Clones this ShapeDecoration object.
	LoadFromXml	Loads the ShapeDecoration definition from an XML document.
	SaveToXml	Saves the ShapeDecoration definition in an XML document.

See Also

ShapeDecoration Class
MindFusion.Diagramming Namespace

ShapeDecoration Constructor

Initializes a new instance of the ShapeDecoration class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeDecoration (
    ElementTemplate[] elements,
    Brush brush,
    FillMode fillMode,
    Pen pen
)
```

Visual Basic Copy Code

```
Public New ( _
    elements() As ElementTemplate, _
    brush As Brush, _
    fillMode As FillMode, _
    pen As Pen _
)
```

Parameters

elements
An array of ElementTemplate objects that define the decoration shape outline.

brush
A Brush object used to fill the decoration shape interior.

fillMode
Specifies how areas are filled if the outline path intersects itself.

pen
A Pen object used to draw the decoration shape outline.

See Also

ShapeDecoration Class
MindFusion.Diagramming Namespace

ShapeDecoration Methods

ShapeDecoration.Clone Method

Clones this ShapeDecoration object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As Object
```

Return Value

A copy of the ShapeDecoration.

See Also

ShapeDecoration Members
ShapeDecoration Class
MindFusion.Diagramming Namespace

ShapeDecoration.LoadFromXml Method

Loads the ShapeDecoration definition from an XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

The parent XmlElement object.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ShapeDecoration Members
ShapeDecoration Class
MindFusion.Diagramming Namespace

ShapeDecoration.SaveToXml Method

Saves the ShapeDecoration definition in an XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
The parent XmlElement object.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ShapeDecoration Members
ShapeDecoration Class
MindFusion.Diagramming Namespace

ShapeDecoration Properties

ShapeDecoration.Brush Property

Gets or sets the brush used to fill the decoration shape interior.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush Brush { get; set; }
```

Visual Basic Copy Code

```
Public Property Brush As Brush
```

Property Value

See Also

ShapeDecoration Members
ShapeDecoration Class
MindFusion.Diagramming Namespace

ShapeDecoration.Elements Property

Gets the elements that define the decoration's shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ElementTemplate[] Elements { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Elements As ElementTemplate()
```

Property Value

See Also

ShapeDecoration Members
ShapeDecoration Class
MindFusion.Diagramming Namespace

ShapeDecoration.FillMode Property

Gets or sets how areas are filled if the outline path intersects itself.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public FillMode FillMode { get; set; }
```

Visual Basic Copy Code

```
Public Property FillMode As FillMode
```

Property Value

See Also

ShapeDecoration Members
ShapeDecoration Class
MindFusion.Diagramming Namespace

ShapeDecoration.Pen Property

Gets or sets the brush used to draw the decoration shape outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen Pen { get; set; }
```

Visual Basic Copy Code

```
Public Property Pen As Pen
```

Property Value

See Also

ShapeDecoration Members
ShapeDecoration Class
MindFusion.Diagramming Namespace

ShapeEventArgs Class

Contains the arguments passed to the **ShapeSelected** and **ShapeRenamed** event handlers.

Syntax

C# Copy Code

```
public class ShapeEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class ShapeEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.ShapeEventArgs

See Also

ShapeEventArgs Members
MindFusion.Diagramming Namespace

ShapeEventArgs Members

Contains the arguments passed to the **ShapeSelected** and **ShapeRenamed** event handlers.

The following tables list the members exposed by the ShapeEventArgs type.

Public Constructors

	Name	Description
	ShapeEventArgs	Overloaded. Initializes a new instance of the **ShapeEventArgs** class.

Public Properties

	Name	Description
	Index	Gets or sets the index of the Shape item within its containing collection.
	OldName	Gets or sets the name of the shape before renaming.
	Shape	Gets the shape related to the event.
	UnknownId	Gets or sets the identifier of a missing shape.

Public Methods

	Name	Description
	ToString	**Object.ToString** override.

See Also

ShapeEventArgs Class
MindFusion.Diagramming Namespace

ShapeEventArgs Constructor

Initializes a new instance of the ShapeEventArgs class.

Overload List

Name	Description
ShapeEventArgs (Shape)	Initializes a new instance of the **ShapeEventArgs** class.
ShapeEventArgs (Shape, Int32)	Initializes a new instance of the **ShapeEventArgs** class.
ShapeEventArgs (Shape, Int32, String)	Initializes a new instance of the **ShapeEventArgs** class.

See Also

ShapeEventArgs Class
MindFusion.Diagramming Namespace

ShapeEventArgs Constructor (Shape)

Initializes a new instance of the ShapeEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeEventArgs (
    Shape shapeTemplate
)
```

Visual Basic Copy Code

```
Public New ( _
    shapeTemplate As Shape _
)
```

Parameters

shapeTemplate
The Shape related to the event.

See Also

ShapeEventArgs Constructor Overload List
ShapeEventArgs Class
MindFusion.Diagramming Namespace

ShapeEventArgs Constructor (Shape, Int32)

Initializes a new instance of the ShapeEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeEventArgs (
    Shape shapeTemplate,
    int index
)
```

Visual Basic Copy Code

```
Public New ( _
    shapeTemplate As Shape, _
    index As Integer _
)
```

Parameters

shapeTemplate
The Shape related to the event.

index
The zero-based index of the shape within its containing collection.

See Also

ShapeEventArgs Constructor Overload List
ShapeEventArgs Class
MindFusion.Diagramming Namespace

ShapeEventArgs Constructor (Shape, Int32, String)

Initializes a new instance of the ShapeEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeEventArgs (
    Shape shapeTemplate,
    int index,
    string oldName
)
```

Visual Basic Copy Code

```
Public New ( _
    shapeTemplate As Shape, _
    index As Integer, _
```

```
    oldName As String _
)
```

Parameters

shapeTemplate
The Shape related to the event.

index
The zero-based index of the shape within its containing collection.

oldName
The old DisplayName of the shape.

See Also

ShapeEventArgs Constructor Overload List
ShapeEventArgs Class
MindFusion.Diagramming Namespace

ShapeEventArgs Methods

ShapeEventArgs.ToString Method

Object.ToString override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override string ToString ()
```

Visual Basic Copy Code

```
Public Overrides Function ToString () As String
```

Return Value

A string that represents the object.

See Also

ShapeEventArgs Members
ShapeEventArgs Class
MindFusion.Diagramming Namespace

ShapeEventArgs Properties

ShapeEventArgs.Index Property

Gets or sets the index of the Shape item within its containing collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Index { get; set; }
```

Visual Basic Copy Code

```
Public Property Index As Integer
```

Property Value

An integer value specifying the zero-based index of the shape.

See Also

ShapeEventArgs Members
ShapeEventArgs Class
MindFusion.Diagramming Namespace

ShapeEventArgs.OldName Property

Gets or sets the name of the shape before renaming.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string OldName { get; set; }
```

Visual Basic Copy Code

```
Public Property OldName As String
```

Property Value

A string specifying the old value of the shape's DisplayName property.

Remarks

This property is relevant only for the **ShapeRenamed** event.

See Also

ShapeEventArgs Members
ShapeEventArgs Class
MindFusion.Diagramming Namespace

ShapeEventArgs.Shape Property

Gets the shape related to the event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape Shape { get; set; }
```

Visual Basic Copy Code

```
Public Property Shape As Shape
```

Property Value

An instance of the Shape class.

See Also

ShapeEventArgs Members
ShapeEventArgs Class
MindFusion.Diagramming Namespace

ShapeEventArgs.UnknownId Property

Gets or sets the identifier of a missing shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string UnknownId { get; set; }
```

Visual Basic Copy Code

```
Public Property UnknownId As String
```

Property Value

Remarks

This property is relevant only for ShapeNotDefined events.

See Also

ShapeEventArgs Members
ShapeEventArgs Class
MindFusion.Diagramming Namespace

ShapeFormulaParam Class

Represents the position of a ShapeControlPoint, passed as parameter to shape formulas.

Syntax

C# Copy Code

```
public class ShapeFormulaParam
```

Visual Basic Copy Code

```
Public Class ShapeFormulaParam
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ShapeFormulaParam

See Also

ShapeFormulaParam Members
MindFusion.Diagramming Namespace

ShapeFormulaParam Members

Represents the position of a ShapeControlPoint, passed as parameter to shape formulas.

The following tables list the members exposed by the ShapeFormulaParam type.

Public Constructors

	Name	Description
	ShapeFormulaParam	Overloaded.

Public Properties

	Name	Description
	X	Gets or sets the horizontal position of the control point.
	Y	Gets or sets the vertical position of the control point.

Public Methods

	Name	Description
	Clone	Creates a copy of this **ShapeFormulaParam**.
	LoadFromXml	Loads the formula parameter from the specified XML element.
	SaveToXml	Saves the formula parameter into the specified XML element.

See Also

ShapeFormulaParam Class
MindFusion.Diagramming Namespace

ShapeFormulaParam Constructor

Overload List

Name	Description
ShapeFormulaParam ()	Initializes a new instance of the ShapeFormulaParam class.
ShapeFormulaParam (Double, Double)	Initializes a new instance of the **ShapeFormulaParam** class.

See Also

ShapeFormulaParam Class
MindFusion.Diagramming Namespace

ShapeFormulaParam Constructor ()

Initializes a new instance of the ShapeFormulaParam class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeFormulaParam ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ShapeFormulaParam Constructor Overload List
ShapeFormulaParam Class
MindFusion.Diagramming Namespace

ShapeFormulaParam Constructor (Double, Double)

Initializes a new instance of the ShapeFormulaParam class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeFormulaParam (
    double x,
    double y
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Double, _
    y As Double _
)
```

Parameters

x
Horizontal position of the control point.

y
Vertical position of the control point.

See Also

ShapeFormulaParam Constructor Overload List
ShapeFormulaParam Class
MindFusion.Diagramming Namespace

ShapeFormulaParam Methods

ShapeFormulaParam.Clone Method

Creates a copy of this ShapeFormulaParam.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeFormulaParam Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As ShapeFormulaParam
```

Return Value

A new ShapeFormulaParam object that is a copy of the current instance.

See Also

ShapeFormulaParam Members
ShapeFormulaParam Class
MindFusion.Diagramming Namespace

ShapeFormulaParam.LoadFromXml Method

Loads the formula parameter from the specified XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** instance.

context
An XmlPersistContext instance.

See Also

ShapeFormulaParam Members
ShapeFormulaParam Class
MindFusion.Diagramming Namespace

ShapeFormulaParam.SaveToXml Method

Saves the formula parameter into the specified XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Public Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** instance.

context

An XmlPersistContext instance.

See Also

ShapeFormulaParam Members
ShapeFormulaParam Class
MindFusion.Diagramming Namespace

ShapeFormulaParam Properties

ShapeFormulaParam.X Property

Gets or sets the horizontal position of the control point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double X { get; set; }
```

Visual Basic Copy Code

```
Public Property X As Double
```

Property Value

A **double** value specifying the horizontal position of the shape control point.

See Also

ShapeFormulaParam Members
ShapeFormulaParam Class
MindFusion.Diagramming Namespace

ShapeFormulaParam.Y Property

Gets or sets the vertical position of the control point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Y { get; set; }
```

Visual Basic Copy Code

```
Public Property Y As Double
```

Property Value

A **double** value specifying the vertical position of the shape control point.

See Also

ShapeLibrary Class

The **ShapeLibrary** class represents a collection of shape definitions stored together.

Syntax

C# Copy Code

```
public class ShapeLibrary
```

Visual Basic Copy Code

```
Public Class ShapeLibrary
```

Remarks

Shape libraries can be created using the **ShapeDesigner** tool that comes with MvcDiagram. Use the LoadFrom method to load a shape library from a file. The loaded shapes can be accessed through the Shapes property. They will be automatically added to the shared list of shapes kept in the Shape class, which lets you access them through the FromId method. A shape library can be saved into a file in XML based format using the SaveToXml method.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.ShapeLibrary

See Also

ShapeLibrary Members
MindFusion.Diagramming Namespace

ShapeLibrary Members

The ShapeLibrary class represents a collection of shape definitions stored together.

The following tables list the members exposed by the **ShapeLibrary** type.

Public Constructors

	Name	Description
	ShapeLibrary	Overloaded.

Public Properties

	Name	Description
	Shapes	Gets or sets the array of all shapes in the library.

Public Methods

	Name	Description
	LoadFrom	Loads a shape library from the specified file.
	LoadFromXml	Overloaded.
	SaveToXml	Overloaded.

See Also

ShapeLibrary Constructor

Overload List

Name	Description
ShapeLibrary ()	Initializes a new empty instance of the ShapeLibrary class.
ShapeLibrary (Shape[])	Initializes a new instance of the ShapeLibrary class.

See Also

ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary Constructor ()

Initializes a new empty instance of the ShapeLibrary class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeLibrary ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ShapeLibrary Constructor Overload List
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary Constructor (Shape[])

Initializes a new instance of the ShapeLibrary class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeLibrary (
    Shape[] shapes
)
```

Visual Basic Copy Code

```
Public New ( _
    shapes() As Shape _
)
```

Parameters

shapes
An array of Shape object to initialize the library with.

See Also

ShapeLibrary Constructor Overload List
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary Methods

ShapeLibrary.LoadFrom Method

Loads a shape library from the specified file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static ShapeLibrary LoadFrom (
    string fileName
)
```

Visual Basic Copy Code

```
Public Shared Function LoadFrom( _
    fileName As String _
) As ShapeLibrary
```

Parameters

fileName
The name of the file that contains the shape library.

Return Value

An instance of the ShapeLibrary class whose Shapes array contains the loaded shape definitions.

See Also

ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary.LoadFromXml Method

Overload List

Name	Description
ShapeLibrary.LoadFromXml (String)	Loads this library from the specified file.
ShapeLibrary.LoadFromXml (XmlDocument)	Loads the shape library from the specified XML document.
ShapeLibrary.LoadFromXml (XmlDocument, String)	Loads the library from the specified XML document. All locations of images associated with shapes in the library are loaded relative to the specified library path.

See Also

ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary.LoadFromXml Method (String)

Loads this library from the specified file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    string fileName
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    fileName As String _
)
```

Parameters

fileName
The name of the file from which to load the shape library.

See Also

LoadFromXml Method Overload List
ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary.LoadFromXml Method (XmlDocument)

Loads the shape library from the specified XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    XmlDocument document
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    document As XmlDocument _
)
```

Parameters

document
The document to load the library from.

See Also

LoadFromXml Method Overload List
ShapeLibrary Members

ShapeLibrary.LoadFromXml Method (XmlDocument, String)

Loads the library from the specified XML document. All locations of images associated with shapes in the library are loaded relative to the specified library path.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void LoadFromXml (
    XmlDocument document,
    string libraryPath
)
```

Visual Basic Copy Code

```
Public Sub LoadFromXml( _
    document As XmlDocument, _
    libraryPath As String _
)
```

Parameters

document

libraryPath

Remarks

If the path is an empty string or null, image locations are loaded relative to the BaseURI of the specified document.

See Also

LoadFromXml Method Overload List
ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary.SaveToXml Method

Overload List

Name	Description
ShapeLibrary.SaveToXml (String)	Saves this library to the specified file.
ShapeLibrary.SaveToXml (XmlDocument)	Saves the shape library into the specified XML document.
ShapeLibrary.SaveToXml (XmlDocument, String)	Saves the shape library into the specified XML document. All locations of images associated with shapes in the library are saved relative to the specified destination path.

See Also

ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary.SaveToXml Method (String)

Saves this library to the specified file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToXml (
    string fileName
)
```

Visual Basic Copy Code

```
Public Sub SaveToXml( _
    fileName As String _
)
```

Parameters

fileName
The name of the file where to save the shape library.

See Also

SaveToXml Method Overload List
ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary.SaveToXml Method (XmlDocument)

Saves the shape library into the specified XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToXml (
    XmlDocument document
)
```

Visual Basic Copy Code

```
Public Sub SaveToXml( _
    document As XmlDocument _
)
```

Parameters

document

See Also

SaveToXml Method Overload List
ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary.SaveToXml Method (XmlDocument, String)

Saves the shape library into the specified XML document. All locations of images associated with shapes in the library are saved relative to the specified destination path.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void SaveToXml (
    XmlDocument document,
    string libraryFileName
)
```

Visual Basic Copy Code

```
Public Sub SaveToXml( _
    document As XmlDocument, _
    libraryFileName As String _
)
```

Parameters

document

libraryFileName

Remarks

If the path is an empty string or null, image locations are written as absolute.

See Also

SaveToXml Method Overload List
ShapeLibrary Members
ShapeLibrary Class
MindFusion.Diagramming Namespace

ShapeLibrary Properties

ShapeLibrary.Shapes Property

Gets or sets the array of all shapes in the library.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape[] Shapes { get; set; }
```

Visual Basic Copy Code

```
Public Property Shapes As Shape()
```

Property Value

An array of Shape objects containing the shape definitions.

See Also

ShapeLibrary Members
ShapeLibrary Class

MindFusion.Diagramming Namespace

ShapeNode Class

ShapeNode instances are diagram nodes that represent geometric shapes.

Syntax

C# Copy Code

```
public class ShapeNode : DiagramNode
```

Visual Basic Copy Code

```
Public Class ShapeNode
    Inherits DiagramNode
```

Remarks

A **ShapeNode** instance represents nodes in graphs, tree, process diagrams, and so on.

The shape of a node is specified through the Shape property. Shapes can be rotated, either interactively using the rotation handle above the node, or programmatically using the RotationAngle property.

Shape nodes can display images and text. Set an image to be painted inside a node by assigning it to the Image property. The position of the image within the node and its alignment are set by the ImageAlign property. To make a node as big as its image, call ResizeToFitImage.

Text inside a shape node is represented by the Text property. Its attributes can be changed by the TextBrush and TextFormat properties. To get even richer text formatting, set EnableStyledText to **true** and embed HTML-like formatting tags in the raw text. Text can be laid out inside arbitrary polygonal shapes by enabling PolygonalTextLayout. To make a node big enough to display its text without clipping, call ResizeToFitText.

To let users edit a node's text inline, call the **BeginEdit** method of **DiagramView**. Call **EndEdit** to exit in-place editing mode.

Fill and border colors can be specified by setting the Pen and Brush properties. Shape nodes can be transparent as set through the Transparent property.

Shape nodes can have incoming and outgoing DiagramLink objects accessible through the **IncomingLinks** and **OutgoingLinks** properties defined in the DiagramNode class.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.ShapeNode
        MindFusion.Diagramming.SvgNode
        MindFusion.Diagramming.SvgNode

See Also

ShapeNode Members
MindFusion.Diagramming Namespace

ShapeNode Members

ShapeNode instances are diagram nodes that represent geometric shapes.

The following tables list the members exposed by the **ShapeNode** type.

Public Constructors

	Name	Description
	ShapeNode	Overloaded. Initializes a new instance of the **ShapeNode** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowFlip	Gets or sets a value indicating whether the node's shape flips when an edge is dragged over its opposite edge.
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	CustomDraw	Gets or sets the type of custom drawing performed on this shape node.
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled.
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	FlipX	Gets or sets a value indicating whether the node's shape is flipped horizontally.
	FlipY	Gets or sets a value indicating whether the node's shape is flipped vertically.
	Font	Gets or sets the font used to render the node's text.
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)

	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	Image	Gets or sets the image that is displayed in this shape node.
	ImageAlign	Gets or sets the alignment of the image displayed in this shape node.
	ImagePadding	Gets or sets the padding distance between node's borders and **Image**.
	ImageUrl	Gets or sets a URL used to load an image into this node.
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	PlainText	Gets the plain text of this shape node.
	PolygonalTextLayout	Gets or sets a value indicating whether the node's Text is formatted inside non-rectangular outline.
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotateImage	Gets or sets a value indicating whether the **Image** should be rotated when the shape node is rotated.
	RotateText	Gets or sets a value indicating whether the **Text** should be rotated when the shape node is rotated.
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)

	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Shape	Gets or sets a reference to the node's geometric shape definition.
	ShapeOrientation	Gets or sets the angle specifying the orientation with which a shape template is applied to the node bounds.
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed inside the shape node.
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of the text that is displayed inside shape nodes.
	TextPadding	Gets or sets the spacing between the item boundaries and its text.
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Transparent	Gets or sets a value indicating whether this shape node is transparent.
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)

AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
Clone	Creates a copy of this item. (Inherited from DiagramItem.)
Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
ContainsPoint	Overloaded.
CreateConnectionPoint	DiagramNode.CreateConnectionPoint override.
Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
DoModify	For internal use. (Inherited from DiagramItem.)
Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
DrawHandles	DiagramNode.DrawHandles override. Renders the adjustment handles of this item.
DrawInRect	(Inherited from DiagramNode.)
DrawLocal	Draws the node on the specified IGraphics surface.
DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified **IGraphics** surface. (Inherited from DiagramNode.)
DrawShadowLocal	Draws the shadow of this node on the specified **IGraphics** surface.
Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline.
GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
GetOutlinePoly	Node.GetOutlinePoly override.
GetRepaintRect	DiagramNode.GetRepaintRect override.

	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramNode.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point.
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	DiagramItem.OnLoad override.
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	ResizeToFitImage	Makes the shape node as big as the image it displays.
	ResizeToFitText	Makes the shape node big enough to display its text without clipping.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)

	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramNode.AllowModify override. Invoked during interactive modification of items to test whether the operation can be completed in the current context.
	CancelModify	DiagramNode.CancelModify override. Invoked when the interactive modification of this object has been canceled.
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramNode.CompleteModify override. Invoked when the item modification has been completed.
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override.
	CreateState	DiagramNode.CreateState override; creates a ShapeNodeState instance.
	Dispose	Disposes any disposable resources managed by this shape.
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetImageRect	Overloaded. Gets the destination rectangle of the node's image.
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a GraphicsPath object for this node as though it is contained in the specified rectangle.
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object.
	LoadFromXml	Loads the item content from an XML element.
	LoadObsolete	For internal use. (Inherited from DiagramItem.)

	OnAdd	Called when the item is added to a **Diagram**, either by users, programmatically or via undo / redo.
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	Called when the Diagram's MeasureUnit changes.
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	DiagramNode.OnRotationAngleChanged override.
	OnStyleChanged	DiagramItem.OnStyleChanged override.
	OnTextChanged	DiagramItem.OnTextChanged override.
	OnTextFormatChanged	DiagramItem.OnTextFormatChanged override.
	OnUpdateBounds	DiagramNode.OnUpdateBounds override.
	OnZoomChanged	Called when the Diagram's **ZoomFactor** changes.
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramNode.RestoreProperties override.
	RestoreState	DiagramNode.RestoreState override. Restores the state of this diagram item.
	SaveProperties	DiagramNode.SaveProperties override.
	SaveState	DiagramNode.SaveState override. Saves the state of this diagram item.
	SaveToJson	Saves the node as a JSON object.
	SaveToXml	Saves the item content into an XML element.
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)

	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramNode.StartModify override. Invoked when the user starts modifying this item interactively.
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	**DiagramNode.UpdateModify** override. Invoked during interactive modification of this item.

See Also

ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode Constructor

Initializes a new instance of the ShapeNode class.

Overload List

Name	Description
ShapeNode ()	Initializes a new instance of the **ShapeNode** class.
ShapeNode (Diagram)	Initializes a new instance of the **ShapeNode** class with default values supplied form the specified Diagram.
ShapeNode (ShapeNode)	Initializes a new instance of the **ShapeNode** class using the specified node as a prototype.

See Also

ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode Constructor ()

Initializes a new instance of the ShapeNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNode ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ShapeNode Constructor Overload List
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode Constructor (Diagram)

Initializes a new instance of the ShapeNode class with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNode (
    Diagram parent
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram _
)
```

Parameters

parent
The Diagram from which to obtain default values for the node properties.

Remarks

This constructor initializes the ShapeNode properties with the values specified in the respective default-value properties of the Diagram object. For more information see **Initial Values of Item Properties**.

See Also

ShapeNode Constructor Overload List
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode Constructor (ShapeNode)

Initializes a new instance of the ShapeNode class using the specified node as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNode (
    ShapeNode prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As ShapeNode _
)
```

Parameters

prototype

The node to use as a prototype. The properties of the new node are initialized from the property values of this node.

Remarks

This constructor copies the ShapeNode property values from the specified prototype node.

See Also

ShapeNode Constructor Overload List
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode Methods

ShapeNode.AllowModify Method

DiagramNode.AllowModify override. Invoked during interactive modification of items to test whether the operation can be completed in the current context.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override bool AllowModify (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Function AllowModify( _
    current As Point, _
    ist As InteractionState _
) As Boolean
```

Parameters

current

The current mouse pointer position.

ist

An InteractionState object containing more information about the current state of the operation.

Return Value

true to allow the modification of this node; otherwise, **false**.

Remarks

This method is invoked after each mouse movement in order to update the mouse pointer icon, so do not perform any time-consuming operations in its implementation.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.CancelModify Method

DiagramNode.CancelModify override. Invoked when the interactive modification of this object has been canceled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void CancelModify (
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub CancelModify( _
    ist As InteractionState _
)
```

Parameters

ist

An InteractionState object containing more information about the current state of the operation.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.CompleteModify Method

DiagramNode.CompleteModify override. Invoked when the item modification has been completed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void CompleteModify (
    Point end,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub CompleteModify( _
    end As Point, _
    ist As InteractionState _
)
```

Parameters

end

The mouse pointer position.

ist

An InteractionState object containing more information about the current state of the operation.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.ContainsPoint Method

Overload List

Name	Description
ShapeNode.ContainsPoint (Point)	DiagramItem.ContainsPoint override.
ShapeNode.ContainsPoint (Point, Double)	Tests whether the specified point is contained within the outline of this node, relaxed with the specified threshold.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.ContainsPoint Method (Point)

DiagramItem.ContainsPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool ContainsPoint (
    Point diagramPoint
)
```

Visual Basic Copy Code

```
Public Overrides Function ContainsPoint( _
    diagramPoint As Point _
) As Boolean
```

Parameters

diagramPoint

Return Value

See Also

ContainsPoint Method Overload List
ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.ContainsPoint Method (Point, Double)

Tests whether the specified point is contained within the outline of this node, relaxed with the specified threshold.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool ContainsPoint (
    Point point,
    double threshold
)
```

Visual Basic Copy Code

```
Public Overrides Function ContainsPoint( _
    point As Point, _
```

```
    threshold As Double _
) As Boolean
```

Parameters

point

threshold

Return Value

See Also

ContainsPoint Method Overload List
ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.CreateConnectionPoint Method

DiagramNode.CreateConnectionPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override ConnectionPoint CreateConnectionPoint (
    DiagramLink link,
    Point point,
    bool incoming
)
```

Visual Basic Copy Code

```
Public Overrides Function CreateConnectionPoint( _
    link As DiagramLink, _
    point As Point, _
    incoming As Boolean _
) As ConnectionPoint
```

Parameters

link
A DiagramLink that should be connected to this node.

point
The point at which the link should be connected.

incoming
true if link is an incoming link; otherwise, **false**.

Return Value

A ConnectionPoint instance representing the connection.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.CreateProperties Method

DiagramItem.CreateProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemProperties CreateProperties ()
```

Visual Basic Copy Code

```
Protected Overrides Function CreateProperties () As DiagramItemProperties
```

Return Value

A DiagramItemProperties -derived object containing a snapshot of the item's properties.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.CreateState Method

DiagramNode.CreateState override; creates a ShapeNodeState instance.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemState CreateState ()
```

Visual Basic Copy Code

```
Protected Overrides Function CreateState () As DiagramItemState
```

Return Value

A ShapeNodeState instance.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.Dispose Method

Disposes any disposable resources managed by this shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void Dispose (
    bool disposing
)
```

Visual Basic Copy Code

```
Protected Overrides Sub Dispose( _
    disposing As Boolean _
)
```

Parameters

disposing
Specifies whether the method is called explicitly or from the garbage collector.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.DrawHandles Method

DiagramNode.DrawHandles override. Renders the adjustment handles of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawHandles (
    IGraphics graphics,
    HandlesVisualStyle style,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawHandles( _
    graphics As IGraphics, _
    style As HandlesVisualStyle, _
    options As RenderOptions _
)
```

Parameters

graphics
The IGraphics surface on which the handles should be drawn.

style
A HandlesVisualStyle instance containing brush and pen objects that should be used when drawing handles.

options
Provides options for the drawing operation.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.DrawLocal Method

Draws the node on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawLocal (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawLocal( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
An IGraphics object where the node should be rendered.

options
Provides options for the drawing operation.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.DrawShadowLocal Method

Draws the shadow of this node on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawShadowLocal (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawShadowLocal( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
An IGraphics object where the shadow should be rendered.

options
Provides options for the drawing operation.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.GetImageRect Method

Gets the destination rectangle of the node's image.

Overload List

Name	Description
ShapeNode.GetImageRect ()	Gets the image rectangle from the associated Shape template, or the node's bounding rectangle if there is no associated shape template.
ShapeNode.GetImageRect (Boolean)	Gets the image rectangle from the associated shape template or the node's bounding rectangle if there is no associated shape template.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.GetImageRect Method ()

Gets the image rectangle from the associated Shape template, or the node's bounding rectangle if there is no associated shape template.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected Rect GetImageRect ()
```

Visual Basic Copy Code

```
Protected Function GetImageRect () As Rect
```

Return Value

A **Microsoft.Maui.Graphics.Rect** specifying the region inside this node where its associated Image will be rendered.

See Also

GetImageRect Method Overload List
ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.GetImageRect Method (Boolean)

Gets the image rectangle from the associated shape template or the node's bounding rectangle if there is no associated shape template.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected Rect GetImageRect (
    bool inLocal
)
```

Visual Basic Copy Code

```
Protected Function GetImageRect( _
    inLocal As Boolean _
) As Rect
```

Parameters

inLocal

Return Value

A **Microsoft.Maui.Graphics.Rect** specifying the region inside this node where its associated Image will be rendered.

See Also

GetImageRect Method Overload List
ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.GetIntersection Method

Calculates the intersection point between the segment defined by the specified point pair and the node's outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Point GetIntersection (
    Point segmentStart,
    Point segmentEnd
)
```

Visual Basic Copy Code

```
Public Overrides Function GetIntersection( _
    segmentStart As Point, _
    segmentEnd As Point _
) As Point
```

Parameters

segmentStart

The first point of the segment.

segmentEnd

The second point of the segment.

Return Value

A **Microsoft.Maui.Graphics.Point** object representing the intersection point.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.GetOutlinePoly Method

Node.GetOutlinePoly override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override PointCollection GetOutlinePoly (
    bool rendering
)
```

Visual Basic

```
Public Overrides Function GetOutlinePoly( _
    rendering As Boolean _
) As PointCollection
```

Parameters

rendering

Return Value

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.GetRepaintRect Method

DiagramNode.GetRepaintRect override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override Rect GetRepaintRect (
    bool includeConnected
)
```

Visual Basic

```
Public Overrides Function GetRepaintRect( _
    includeConnected As Boolean _
) As Rect
```

Parameters

includeConnected
true if related items should be considered too; otherwise, **false**.

Return Value

A rectangular region that should be repainted when this node is modified.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.GetRgnInRect Method

Creates and returns a GraphicsPath object for this node as though it is contained in the specified rectangle.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override IGraphicsPath GetRgnInRect (
    Rect rc,
    bool inLocal
)
```

Visual Basic Copy Code

```
Protected Overrides Function GetRgnInRect( _
    rc As Rect, _
    inLocal As Boolean _
) As IGraphicsPath
```

Parameters

rc

inLocal

Return Value

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.HitTestHandle Method

DiagramNode.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override AdjustmentHandle HitTestHandle (
    Point point,
    double distance
)
```

Visual Basic Copy Code

```
Public Overrides Function HitTestHandle( _
    point As Point, _
    distance As Double _
) As AdjustmentHandle
```

Parameters

point
A **Microsoft.Maui.Graphics.Point** specifying where to look for a selection handle.

distance
Maximum distance from a handle at which it is still considered as hit.

Return Value

Returns an AdjustmentHandle instance representing the adjustment handle found at the specified location, or **null** if there is none.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.LoadFromJson Method

Loads the node from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the node's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.LoadFromXml Method

Loads the item content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** containing the item's data.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.OnAdd Method

Called when the item is added to a Diagram, either by users, programmatically or via undo / redo.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnAdd ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnAdd ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.OnLoad Method

DiagramItem.OnLoad override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void OnLoad ()
```

Visual Basic Copy Code

```
Public Overrides Sub OnLoad ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.OnMeasureUnitChanged Method

Called when the Diagram's MeasureUnit changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnMeasureUnitChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnMeasureUnitChanged ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.OnRotationAngleChanged Method

DiagramNode.OnRotationAngleChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnRotationAngleChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnRotationAngleChanged ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.OnStyleChanged Method

DiagramItem.OnStyleChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnStyleChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnStyleChanged ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.OnTextChanged Method

DiagramItem.OnTextChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnTextChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnTextChanged ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.OnTextFormatChanged Method

DiagramItem.OnTextFormatChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnTextFormatChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnTextFormatChanged ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.OnUpdateBounds Method

DiagramNode.OnUpdateBounds override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnUpdateBounds (
    bool sizeChanged
)
```

Visual Basic Copy Code

```
Protected Overrides Sub OnUpdateBounds( _
    sizeChanged As Boolean _
)
```

Parameters

sizeChanged
true if the size has changed; **false** if only the position has changed.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.OnZoomChanged Method

Called when the Diagram's **ZoomFactor** changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnZoomChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnZoomChanged ()
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.ResizeToFitImage Method

Makes the shape node as big as the image it displays.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ResizeToFitImage ()
```

Visual Basic Copy Code

```
Public Sub ResizeToFitImage ()
```

Remarks

Call this method to make a shape node exactly the same size as the image it contains. The dots-per-inch resolution of the image is considered too when calculating the new node size. Use the Image property to assign an image to a node and ImageAlign to specify position of the image within the node.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
ResizeToFitText Method

ShapeNode.ResizeToFitText Method

Makes the shape node big enough to display its text without clipping.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ResizeToFitText (
    FitSize fit
)
```

Visual Basic Copy Code

```
Public Function ResizeToFitText( _
    fit As FitSize _
) As Boolean
```

Parameters

fit
A member of the FitSize enumeration specifying whether the width or height of the node should remain unchanged.

Return Value

true if the node is resized successfully, or **false** otherwise.

Example

The following sample resizes a node to fit its text, using 1:1 width to height ratio.

C# Copy Code

```
var r = new Rect(shapeNode.Bounds.Left, shapeNode.Bounds.Top, 10, 10);

shapeNode.PolygonalTextLayout = true;
shapeNode.Bounds = r;
shapeNode.Text = "the method changes both the box' width" +
    "and height, maintaining their original ratio.";
shapeNode.ResizeToFitText(FitSize.KeepRatio);
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
Text Property
ResizeToFitImage Method

ShapeNode.RestoreProperties Method

DiagramNode.RestoreProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties -derived object containing a snapshot of the item's properties.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.RestoreState Method

DiagramNode.RestoreState override. Restores the state of this diagram item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreState (
    DiagramItemState state
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreState( _
    state As DiagramItemState _
)
```

Parameters

state
A ShapeNodeState instance, previously created by CreateState.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.SaveProperties Method

DiagramNode.SaveProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties -derived object where the item's properties should be saved.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.SaveState Method

DiagramNode.SaveState override. Saves the state of this diagram item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveState (
    DiagramItemState state
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveState( _
    state As DiagramItemState _
)
```

Parameters

state
A ShapeNodeState instance previously created by CreateState.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.SaveToJson Method

Saves the node as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject instance to contain the serialized data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.SaveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** the item's data should be stored into.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.StartModify Method

DiagramNode.StartModify override. Invoked when the user starts modifying this item interactively.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void StartModify (
    Point org,
    AdjustmentHandle handle,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub StartModify( _
    org As Point, _
    handle As AdjustmentHandle, _
    ist As InteractionState _
)
```

Parameters

org
Position of the mouse pointer.

handle
The adjustment handle used to modify this node.

ist
An InteractionState object containing more information about the current state of the operation.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.UpdateModify Method

DiagramNode.UpdateModify override. Invoked during interactive modification of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void UpdateModify (
    Point current,
    InteractionState ist
)
```

Visual Basic Copy Code

```
Protected Overrides Sub UpdateModify( _
    current As Point, _
    ist As InteractionState _
)
```

Parameters

current
Position of the mouse pointer.

ist
An InteractionState object containing more information about the current state of the operation.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode Properties

ShapeNode.AllowFlip Property

Gets or sets a value indicating whether the node's shape flips when an edge is dragged over its opposite edge.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowFlip { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowFlip As Boolean
```

Property Value

true to let users flip the shape, or **false** otherwise. The default value is **false**.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
FlipX Property
FlipY Property
EnabledHandles Property
Shape Property

ShapeNode.CustomDraw Property

Gets or sets the type of custom drawing performed on this shape node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CustomDraw CustomDraw { get; set; }
```

Visual Basic Copy Code

```
Public Property CustomDraw As CustomDraw
```

Property Value

A member of the CustomDraw enumeration. Initialized with the value of ShapeCustomDraw.

Remarks

This property allows invoking custom painting code for a shape node. That code must execute in response to the DrawNode event. There are several modes of custom painting, allowing the custom code to paint the shape node entirely, or do that in addition to what MvcDiagram paints by default. The following CustomDraw members can be assigned to this property to specify the custom painting mode:

Member Name	Description
None	Do not perform custom drawing, let all painting be done by MvcDiagram.
Additional	The custom painting code executes after the node interior and frame are painted by MvcDiagram, but before text is rendered. In this mode, the **Graphics** instance passed to the custom drawing event handler has a clipping region applied to it.
Full	The component does not do any standard drawing, the entire painting of the node is done by the application.
ShadowOnly	Only the shadow of the node is painted by custom code, the node itself is drawn by the component.

Additional2	The custom painting code executes after all parts of the node are painted by MvcDiagram. In this mode, the **Graphics** instance passed to the custom drawing event handler doesn't have a clipping region applied to it.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.EnableStyledText Property

Gets or sets a value indicating whether styled text rendering is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableStyledText { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableStyledText As Boolean
```

Property Value

true to parse the node's text in order to find formatting tags; otherwise, **false**. Initialized with the value of EnableStyledText.

Remarks

Styled text allows applying various attributes to the characters it contains. Attributes are specified using HTML-like formatting tags embedded in the raw text. Formatting tags are listed and described in the **Text Attributes and Formatting** section of this help file.

Styled text can be enabled only if PolygonalTextLayout is set to **true**. If enabled, Text is parsed and analyzed each time it is changed. Parsing determines what formatting tags are used and adds the corresponding attribute records to MvcDiagram's internal representation of styled text. If the property value is **false**, the text is not processed in any way and is displayed without formatting.

Example

The following sample code demonstrates how to use styled text:

C# Copy Code

```
shapeNode.PolygonalTextLayout = true;
shapeNode.EnableStyledText = true;
shapeNode.Text = "CH<sub>2</sub>CL<sub>2</sub> " + "\n" +
    "<b><color=#0088FF>methylene-chloride</b>";
```

Visual Basic Copy Code

```
shapeNode.PolygonalTextLayout = True
shapeNode.EnableStyledText = True
shapeNode.Text = "CH<sub>2</sub>CL<sub>2</sub> " + vbNewLine + _
    "<b><color=#0088FF>methylene-chloride</b>"
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
TextFormat Property
Text Attributes and Formatting

ShapeNode.FlipX Property

Gets or sets a value indicating whether the node's shape is flipped horizontally.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool FlipX { get; set; }
```

Visual Basic Copy Code

```
Public Property FlipX As Boolean
```

Property Value

true if the shape is flipped horizontally, or **false** otherwise.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
FlipY Property
AllowFlip Property

ShapeNode.FlipY Property

Gets or sets a value indicating whether the node's shape is flipped vertically.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool FlipY { get; set; }
```

Visual Basic Copy Code

```
Public Property FlipY As Boolean
```

Property Value

true if the shape is flipped vertically, or **false** otherwise.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
FlipX Property
AllowFlip Property

ShapeNode.Font Property

Gets or sets the font used to render the node's text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Font Font { get; set; }
```

Visual Basic Copy Code

```
Public Overrides Property Font As Font
```

Property Value

A Font instance specifying the font attributes.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.Image Property

Gets or sets the image that is displayed in this shape node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage Image { get; set; }
```

Visual Basic Copy Code

```
Public Property Image As IImage
```

Property Value

A **Microsoft.Maui.Graphics.IImage** instance. The default value is **null** (**Nothing** in Visual Basic).

Remarks

This property lets you display an image inside the node. The ImageRectangle attribute of the shape definition specifies a rectangular area inside the node in which the image is displayed. Use the ImageAlign property to specify how the image is aligned to that area. By default, the image is displayed in the center of the bounding rectangle of the node.

If there isn't any image assigned to this property, the shape node displays the Image specified in its Shape template.

If a node should display multiple images, you could add and arrange them using NodeLabel objects.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
RotateImage Property
ResizeToFitImage Method
ImagePadding Property

ShapeNode.ImageAlign Property

Gets or sets the alignment of the image displayed in this shape node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ImageAlign ImageAlign { get; set; }
```

Visual Basic Copy Code

```
Public Property ImageAlign As ImageAlign
```

Property Value

A member of the ImageAlign enumeration. The default is **Center**.

Remarks

This property specifies how the Image is aligned to the bounding rectangle of this node. The image is aligned relatively to the rectangle specified in the ImageRectangle attribute of the Shape definition.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
Image Property
ImagePadding Property

ShapeNode.ImagePadding Property

Gets or sets the padding distance between node's borders and Image.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Nullable<Thickness> ImagePadding { get; set; }
```

Visual Basic Copy Code

```
Public Property ImagePadding As Nullable(Of Thickness)
```

Property Value

An optional Thickness value specifying padding distance; the default value is **null**.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
ImageAlign Property

ShapeNode.ImageUrl Property

Gets or sets a URL used to load an image into this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ImageUrl { get; set; }
```

Visual Basic Copy Code

```
Public Property ImageUrl As String
```

Property Value

A string specifying the image location.

Remarks

This property is used in MindFusion Web components. If it is set, the image data is not transferred to the client-side together with the diagram but only the URL string. Such images are downloaded asynchronously after all other diagram elements load on the client.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
Image Property

ShapeNode.PlainText Property

Gets the plain text of this shape node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string PlainText { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property PlainText As String
```

Property Value

A string containing the plain text of the shape node.

Remarks

This property removes any formatting tags from the node's Text and returns the plain human-readable text. That is useful when styled text formatting is enabled and you need to display the node's text outside of MvcDiagram, for example in a message box or a dialog box.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
EnableStyledText Property

ShapeNode.PolygonalTextLayout Property

Gets or sets a value indicating whether the node's Text is formatted inside non-rectangular outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool PolygonalTextLayout { get; set; }
```

Visual Basic Copy Code

```
Public Property PolygonalTextLayout As Boolean
```

Property Value

true to enable polygonal formatting; **false** to disable it.

Remarks

The node's area in which text is rendered can be explicitly defined for each Shape definition, and if it is not specified, the shape outline is used implicitly. The formatting algorithm is designed to layout text inside polygonal outlines. A polygonal approximation is created for outlines that have curved segments, so text can be laid out in shapes that consist of Bézier curves too.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
Text Property
EnableStyledText Property
Shape Property

ShapeNode.RotateImage Property

Gets or sets a value indicating whether the Image should be rotated when the shape node is rotated.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RotateImage { get; set; }
```

Visual Basic Copy Code

```
Public Property RotateImage As Boolean
```

Property Value

true to rotate the node's image; otherwise, **false**. The default is **true**.

Remarks

The image is clipped by the rotated shape outline, even if this property is set to **false**.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
RotationAngle Property
RotateText Property

ShapeNode.RotateText Property

Gets or sets a value indicating whether the Text should be rotated when the shape node is rotated.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RotateText { get; set; }
```

Visual Basic Copy Code

```
Public Property RotateText As Boolean
```

Property Value

true to rotate the node's text; otherwise, **false**. The default is **false**.

Remarks

If the property is set to **false** and PolygonalTextLayout is enabled, the text stays horizontal but it adapts to the rotated outline of the shape node.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
RotationAngle Property
RotateImage Property

ShapeNode.Shape Property

Gets or sets a reference to the node's geometric shape definition.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape Shape { get; set; }
```

Visual Basic Copy Code

```
Public Property Shape As Shape
```

Property Value

An instance of the Shape class.

Remarks

Use this property to assign a complex shape to a node. Complex shapes consist of outlines, decorations and text-region definitions that can be defined as series of lines, arcs and curves. To define a custom shape, create an instance of the Shape class. That class also provides 87 predefined shapes exposed by the Shapes collection.

Example

The following sample code illustrates how to use the predefined shapes and how to create your own custom shapes.

C# Copy Code

```
private void example_Click(object sender, System.EventArgs e)
{
    ShapeNode shapeNode0 = diagram.Nodes[0] as ShapeNode;
    ShapeNode shapeNode1 = diagram.Nodes[1] as ShapeNode;

    // Use one of the predefined shapes
    shapeNode0.Shape = Shape.FromId("MultiDocument");

    // Create a custom shape
    shapeNode1.Shape = new Shape(
        new ElementTemplate[] {
            new LineTemplate(0, 100, 60, 100),
            new BezierTemplate(60, 100, 80, 100, 100, 75, 100, 50),
            new BezierTemplate(100, 50, 100, 25, 80, 0,  60, 0),
            new LineTemplate(60, 0, 0, 0),
            new LineTemplate(0, 0, 0, 100) },
        FillMode.Winding);
}
```

Visual Basic Copy Code

```
Private Sub example_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
example.Click

    Dim shapeNode0 As ShapeNode = CType(diagram.Nodes(0), ShapeNode)
    Dim shapeNode1 As ShapeNode = CType(diagram.Nodes(1), ShapeNode)

    ' Use one of the predefined shapes
    shapeNode0.Shape = Shape.FromId("MultiDocument")

    ' Create a custom shape
    shapeNode1.Shape = New Shape( _
        New ElementTemplate() { _
            New LineTemplate(0, 100, 60, 100), _
            New BezierTemplate(60, 100, 80, 100, 100, 75, 100, 50), _
            New BezierTemplate(100, 50, 100, 25, 80, 0, 60, 0), _
            New LineTemplate(60, 0, 0, 0), _
            New LineTemplate(0, 0, 0, 100)}, _
        FillMode.Winding)

End Sub
```

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
Shape.FromId Method
Shape.Shapes Property
PolygonalTextLayout Property
ShapeOrientation Property
AllowFlip Property

ShapeNode.ShapeOrientation Property

Gets or sets the angle specifying the orientation with which a shape template is applied to the node bounds.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ShapeOrientation { get; set; }
```

Visual Basic Copy Code

```
Public Property ShapeOrientation As Double
```

Property Value

A **double** value specifying the orientation angle in degrees. Initialized with the value of ShapeOrientation.

Remarks

That property does not rotate the node, that is, does not rotate the bounding rectangle of the node around its center. It specifies a rotation transformation to be performed on the definition of the Shape when it is applied to a node; the node's bounding rectangle (location, width and height) remain intact. In the current MvcDiagram version, only 0, 90, 180 or 270 can be set as value of **ShapeOrientation**.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
Shape Property
RotationAngle Property

ShapeNode.Text Property

Gets or sets the text displayed inside the shape node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Text { get; set; }
```

Visual Basic Copy Code

```
Public Property Text As String
```

Property Value

A string containing the node's text. Initialized with the value of ShapeText.

Remarks

This property specifies the text that should be drawn inside the shape. The text appearance can be customized through the TextBrush and TextFormat properties.

Styled text allows applying various attributes to the text characters. Attributes are specified by embedding HTML-like formatting tags within the raw text. Styled text is enabled by setting EnableStyledText to **true**.

You can add more text labels by calling the AddLabel method, and customize their position and appearance via the various members of returned NodeLabel object.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
Font Property
Labels Property
ToolTip Property

ShapeNode.TextFormat Property

Gets or sets the alignment of the text that is displayed inside shape nodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringFormat TextFormat { get; set; }
```

Visual Basic Copy Code

```
Public Property TextFormat As StringFormat
```

Property Value

An instance of the StringFormat class. Initialized with the value of TextFormat.

Remarks

Use this property to specify the wrapping mode and alignment of the text that is contained in nodes.

If there isn't enough space available to fit a whole line of text, MvcDiagram wraps the text to a new line when it reaches the edge of the node. You can specify that the text should be wrapped only at word boundaries by setting the **Trimming** member of **TextFormat** to **StringTrimming.Word**. The default is to split words into multiple parts if there is not enough space to fit a whole word.

If you prefer to use your own line-breaks, you can disable the automatic text wrapping by enabling the **NoWrap** flag in the **FormatFlags** member of **TextFormat**. If automatic wrapping is disabled and there isn't enough space to fit the whole line, the text is clipped at the node boundaries.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace
Text Property
TextColor Property
Diagram.TextFormat Property

ShapeNode.TextPadding Property

Gets or sets the spacing between the item boundaries and its text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Thickness TextPadding { get; set; }
```

Visual Basic Copy Code

```
Public Overrides Property TextPadding As Thickness
```

Property Value

A Thickness instance specifying the size of the padding space left around the text.

See Also

ShapeNode Members
ShapeNode Class
MindFusion.Diagramming Namespace

ShapeNode.Transparent Property

Gets or sets a value indicating whether this shape node is transparent.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Transparent { get; set; }
```

Visual Basic Copy Code

```
Public Property Transparent As Boolean
```

Property Value

true if the shape node is transparent; otherwise, **false**. The default is **false**.

Remarks

If a node is transparent, only its image and text are drawn. Shape, filling and frames are not painted.

See Also

ShapeNodeProperties Class

Stores the values of ShapeNode properties during undo / redo operations.

Syntax

C# Copy Code

```
public class ShapeNodeProperties : DiagramNodeProperties
```

Visual Basic Copy Code

```
Public Class ShapeNodeProperties
    Inherits DiagramNodeProperties
```

Remarks

If you need to implement undo support for the additional properties defined by a class derived from ShapeNode, do so by deriving from **ShapeNodeProperties** and overriding the SaveProperties, RestoreProperties, and CreateProperties methods of **ShapeNode**.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItemProperties
    MindFusion.Diagramming.DiagramNodeProperties
      MindFusion.Diagramming.ShapeNodeProperties
        MindFusion.Diagramming.SvgNodeProperties

See Also

ShapeNodeProperties Members
MindFusion.Diagramming Namespace

ShapeNodeProperties Members

Stores the values of ShapeNode properties during undo / redo operations.

The following tables list the members exposed by the ShapeNodeProperties type.

Public Constructors

	Name	Description
	ShapeNodeProperties	Initializes a new instance of the **ShapeNodeProperties** class.

Public Fields

	Name	Description
	AllowFlip	For internal use.
	FlipX	For internal use.
	FlipY	For internal use.
	ItemClass	For internal use. (Inherited from DiagramItemProperties.)
	labels	For internal use. (Inherited from DiagramNodeProperties.)
	RotateText	For internal use.

	Selected	For internal use. (Inherited from DiagramItemProperties.)
	Shape	For internal use.
	ShapeOrientation	For internal use.
	Text	For internal use.
	TextPadding	For internal use. (Inherited from DiagramItemProperties.)
	Transparent	For internal use.
	UsePolyTextLt	For internal use.
	ZIndex	For internal use. (Inherited from DiagramItemProperties.)

Public Methods

	Name	Description
	FreeResources	DiagramItemProperties.FreeResources override.

See Also

ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeProperties Fields

ShapeNodeProperties.AllowFlip Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowFlip
```

Visual Basic Copy Code

```
Public AllowFlip As Boolean
```

See Also

ShapeNodeProperties Members
ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeProperties.FlipX Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool FlipX
```

Visual Basic Copy Code

```
Public FlipX As Boolean
```

See Also

ShapeNodeProperties Members
ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeProperties.FlipY Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool FlipY
```

Visual Basic Copy Code

```
Public FlipY As Boolean
```

See Also

ShapeNodeProperties Members
ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeProperties.RotateText Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool RotateText
```

Visual Basic Copy Code

```
Public RotateText As Boolean
```

See Also

ShapeNodeProperties Members
ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeProperties.Shape Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape Shape
```

Visual Basic Copy Code

```
Public Shape As Shape
```

See Also

ShapeNodeProperties Members
ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeProperties.ShapeOrientation Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ShapeOrientation
```

Visual Basic Copy Code

```
Public ShapeOrientation As Double
```

See Also

ShapeNodeProperties Members
ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeProperties.Text Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Text
```

Visual Basic Copy Code

```
Public Text As String
```

See Also

ShapeNodeProperties Members
ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeProperties.Transparent Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Transparent
```

Visual Basic Copy Code

```
Public Transparent As Boolean
```

See Also

ShapeNodeProperties Members
ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeProperties.UsePolyTextLt Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool UsePolyTextLt
```

Visual Basic Copy Code

```
Public UsePolyTextLt As Boolean
```

See Also

ShapeNodeProperties Members
ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeProperties Constructor

Initializes a new instance of the ShapeNodeProperties class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNodeProperties ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeProperties Methods

ShapeNodeProperties.FreeResources Method

DiagramItemProperties.FreeResources override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void FreeResources ()
```

Visual Basic Copy Code

```
Public Overrides Sub FreeResources ()
```

See Also

ShapeNodeProperties Members
ShapeNodeProperties Class
MindFusion.Diagramming Namespace

ShapeNodeState Class

Contains a snapshot of the internal state of a ShapeNode. Instances of this class are used to store the state of nodes for later undo or redo operations. If you derive from **ShapeNode** and need to implement undo support for the additional internal state defined by your class, do so by deriving from **ShapeNodeState** and overriding the ShapeNode.CreateState, ShapeNode.SaveState and ShapeNode.RestoreState methods.

Syntax

C# Copy Code

```
public class ShapeNodeState : DiagramNodeState
```

Visual Basic Copy Code

```
Public Class ShapeNodeState
    Inherits DiagramNodeState
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItemState
    MindFusion.Diagramming.DiagramNodeState
      MindFusion.Diagramming.ShapeNodeState

See Also

ShapeNodeState Members
MindFusion.Diagramming Namespace

ShapeNodeState Members

Contains a snapshot of the internal state of a ShapeNode. Instances of this class are used to store the state of nodes for later undo or redo operations. If you derive from **ShapeNode** and need to implement undo support for the additional internal state defined by your class, do so by deriving from ShapeNodeState and overriding the ShapeNode.CreateState, ShapeNode.SaveState and ShapeNode.RestoreState methods.

The following tables list the members exposed by the **ShapeNodeState** type.

Public Constructors

	Name	Description
	ShapeNodeState	Initializes a new instance of the **ShapeNodeState** class.

See Also

ShapeNodeState Class
MindFusion.Diagramming Namespace

ShapeNodeState Constructor

Initializes a new instance of the ShapeNodeState class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNodeState ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ShapeNodeState Class
MindFusion.Diagramming Namespace

ShapeNodeStyle Class

Represents a reusable set of properties that define the appearance of shape nodes.

Syntax

C# Copy Code

```
public class ShapeNodeStyle : DiagramNodeStyle
```

Visual Basic Copy Code

```
Public Class ShapeNodeStyle
    Inherits DiagramNodeStyle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle
      MindFusion.Diagramming.DiagramNodeStyle
        MindFusion.Diagramming.ShapeNodeStyle

See Also

ShapeNodeStyle Members
MindFusion.Diagramming Namespace

ShapeNodeStyle Members

Represents a reusable set of properties that define the appearance of shape nodes.

The following tables list the members exposed by the ShapeNodeStyle type.

Public Constructors

	Name	Description
	ShapeNodeStyle	Initializes a new instance of the **ShapeNodeStyle** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of the item. (Inherited from DiagramItemStyle.)
	Effects	Gets a list of the effects applied to the node. (Inherited from DiagramNodeStyle.)
	FontFamily	Gets or sets the font family of the item text. (Inherited from DiagramItemStyle.)
	FontSize	Gets or sets the font size of the item text. (Inherited from DiagramItemStyle.)
	FontStyle	Gets or sets the font style of the item text. (Inherited from DiagramItemStyle.)
	FontUnit	Gets or sets the font unit of the item text. (Inherited from DiagramItemStyle.)
	ShadowBrush	Gets or sets the brush used to paint the shadow of the item. (Inherited from DiagramItemStyle.)
	Stroke	Gets or sets the brush used to paint the outline of the item. (Inherited from DiagramItemStyle.)
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items. (Inherited from DiagramItemStyle.)
	StrokeThickness	Gets or sets the thickness of the item's outline, expressed in the current units of measure. (Inherited from DiagramItemStyle.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItemStyle.)

Public Methods

	Name	Description
	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description

	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JsonObject. (Inherited from DiagramNodeStyle.)
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)
	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a JsonObject. (Inherited from DiagramNodeStyle.)
	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

ShapeNodeStyle Class
MindFusion.Diagramming Namespace

ShapeNodeStyle Constructor

Initializes a new instance of the ShapeNodeStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ShapeNodeStyle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ShapeNodeStyle Class
MindFusion.Diagramming Namespace

ShapeNodeStyle Events

Shapes Class

Provides alternative method of access to the MvcDiagram **predefined shapes**, that allows code completion and IntelliSense support, which is not available when using Shape.FromId.

Syntax

C# Copy Code

```
public class Shapes
```

Visual Basic Copy Code

```
Public Class Shapes
```

Remarks

The accessor properties are generating using the following code:

C# Copy Code

```
TextWriter writer = new StreamWriter(@"D:\shapes.txt");
foreach (Shape shape in Shape.Shapes)
{
    writer.WriteLine("static public Shape " + shape.Id);
    writer.WriteLine("{");
    writer.WriteLine("\tget { return Shape.FromId(\"" + shape.Id + "\"); }");
    writer.WriteLine("}");
    writer.WriteLine();
    writer.Flush();
}
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Shapes

See Also

Shapes Members
MindFusion.Diagramming Namespace
Table of Predefined Shapes

Shapes Members

Provides alternative method of access to the MvcDiagram **predefined shapes**, that allows code completion and IntelliSense support, which is not available when using Shape.FromId.

The following tables list the members exposed by the Shapes type.

Public Properties

	Name	Description
	Actor	Gets the Actor predefined shape.
	Alternative	Gets the Alternative predefined shape.
	Arrow1	Gets the Arrow1 predefined shape.
	Arrow2	Gets the Arrow2 predefined shape.
	Arrow3	Gets the Arrow3 predefined shape.
	Arrow4	Gets the Arrow4 predefined shape.
	Arrow5	Gets the Arrow5 predefined shape.
	Arrow6	Gets the Arrow6 predefined shape.
	Arrow7	Gets the Arrow7 predefined shape.
	Arrow8	Gets the Arrow8 predefined shape.

	Arrow9	Gets the Arrow9 predefined shape.
	BeginLoop	Gets the BeginLoop predefined shape.
	BpmnComplex	Gets the BpmnComplex predefined shape.
	BpmnDataBasedXor	Gets the BpmnDataBasedXor predefined shape.
	BpmnEndCancel	Gets the BpmnEndCancel predefined shape.
	BpmnEndCompensation	Gets the BpmnEndCompensation predefined shape.
	BpmnEndError	Gets the BpmnEndError predefined shape.
	BpmnEndLink	Gets the BpmnEndLink predefined shape.
	BpmnEndMessage	Gets the BpmnEndMessage predefined shape.
	BpmnEndMultiple	Gets the BpmnEndMultiple predefined shape.
	BpmnEndTerminate	Gets the BpmnEndTerminate predefined shape.
	BpmnEventBasedXor	Gets the BpmnEventBasedXor predefined shape.
	BpmnInclusive	Gets the BpmnInclusive predefined shape.
	BpmnIntermediateCancel	Gets the BpmnIntermediateCancel predefined shape.
	BpmnIntermediateCompensation	Gets the BpmnIntermediateCompensation predefined shape.
	BpmnIntermediateError	Gets the BpmnIntermediateError predefined shape.
	BpmnIntermediateLink	Gets the BpmnIntermediateLink predefined shape.
	BpmnIntermediateMessage	Gets the BpmnIntermediateMessage predefined shape.
	BpmnIntermediateMultiple	Gets the BpmnIntermediateMultiple predefined shape.
	BpmnIntermediateRule	Gets the BpmnIntermediateRule predefined shape.
	BpmnIntermediateTimer	Gets the BpmnIntermediateTimer predefined shape.
	BpmnParallelFork	Gets the BpmnParallelFork predefined shape.
	BpmnStartLink	Gets the BpmnStartLink predefined shape.
	BpmnStartMessage	Gets the BpmnStartMessage predefined shape.
	BpmnStartMultiple	Gets the BpmnStartMultiple predefined shape.
	BpmnStartRule	Gets the BpmnStartRule predefined shape.
	BpmnStartTimer	Gets the BpmnStartTimer predefined shape.
	Cloud	Gets the Cloud predefined shape.
	Collate	Gets the Collate predefined shape.
	ConeDown	Gets the ConeDown predefined shape.
	ConeUp	Gets the ConeUp predefined shape.
	ConnectedIssues	Gets the ConnectedIssues predefined shape.
	CreateRequest	Gets the CreateRequest predefined shape.
	Cross	Gets the Cross predefined shape.
	Cube	Gets the Cube predefined shape.
	Cylinder	Gets the Cylinder predefined shape.
	Database	Gets the Database predefined shape.

	DataTransmition	Gets the DataTransmition predefined shape.
	DDelay	Gets the DDelay predefined shape.
	Decagon	Gets the Decagon predefined shape.
	Decision	Gets the Decision predefined shape.
	Decision2	Gets the Decision2 predefined shape.
	Delay	Gets the Delay predefined shape.
	DInDelay	Gets the DInDelay predefined shape.
	DirectAccessStorage	Gets the DirectAccessStorage predefined shape.
	DiskStorage	Gets the DiskStorage predefined shape.
	Display	Gets the Display predefined shape.
	DividedEvent	Gets the DividedEvent predefined shape.
	DividedProcess	Gets the DividedProcess predefined shape.
	Document	Gets the Document predefined shape.
	Donut	Gets the Donut predefined shape.
	DOutDelay	Gets the DOutDelay predefined shape.
	Ellipse	Gets the Ellipse predefined shape.
	EndLoop	Gets the EndLoop predefined shape.
	ExternalOrganization	Gets the ExternalOrganization predefined shape.
	ExternalProcess	Gets the ExternalProcess predefined shape.
	File	Gets the File predefined shape.
	FramedRectangle	Gets the FramedRectangle predefined shape.
	Gate	Gets the Gate predefined shape.
	Heart	Gets the Heart predefined shape.
	Heptagon	Gets the Heptagon predefined shape.
	Hourglass	Gets the Hourglass predefined shape.
	Input	Gets the Input predefined shape.
	InternalStorage	Gets the InternalStorage predefined shape.
	Interrupt	Gets the Interrupt predefined shape.
	Interrupt2	Gets the Interrupt2 predefined shape.
	IsoProcess	Gets the IsoProcess predefined shape.
	Junction	Gets the Junction predefined shape.
	Lightning	Gets the Lightning predefined shape.
	LinedDocument	Gets the LinedDocument predefined shape.
	ManualOperation	Gets the ManualOperation predefined shape.
	Merge	Gets the Merge predefined shape.
	MessageFromUser	Gets the MessageFromUser predefined shape.
	MessageToUser	Gets the MessageToUser predefined shape.

	Microform	Gets the Microform predefined shape.
	MicroformProcessing	Gets the MicroformProcessing predefined shape.
	MicroformRecording	Gets the MicroformRecording predefined shape.
	MultiDocument	Gets the MultiDocument predefined shape.
	MultiProc	Gets the MultiProc predefined shape.
	Octagon	Gets the Octagon predefined shape.
	OfflineStorage	Gets the OfflineStorage predefined shape.
	OffpageConnection	Gets the OffpageConnection predefined shape.
	OffpageReference	Gets the OffpageReference predefined shape.
	Or	Gets the Or predefined shape.
	Pentagon	Gets the Pentagon predefined shape.
	Plaque	Gets the Plaque predefined shape.
	PrimitiveFromCall	Gets the PrimitiveFromCall predefined shape.
	PrimitiveToCall	Gets the PrimitiveToCall predefined shape.
	Procedure	Gets the Procedure predefined shape.
	Process	Gets the Process predefined shape.
	Prysm	Gets the Prysm predefined shape.
	PunchedCard	Gets the PunchedCard predefined shape.
	PyramidDown	Gets the PyramidDown predefined shape.
	PyramidUp	Gets the PyramidUp predefined shape.
	Rectangle	Gets the Rectangle predefined shape.
	RightTriangle	Gets the RightTriangle predefined shape.
	RoundRect	Gets the RoundRect predefined shape.
	RSave	Gets the RSave predefined shape.
	Save	Gets the Save predefined shape.
	Sort	Gets the Sort predefined shape.
	Sphere	Gets the Sphere predefined shape.
	Star	Gets the Star predefined shape.
	Star16Pointed	Gets the Star16Pointed predefined shape.
	Star24Pointed	Gets the Star24Pointed predefined shape.
	Star32Pointed	Gets the Star32Pointed predefined shape.
	Star4Pointed	Gets the Star4Pointed predefined shape.
	Star5Pointed	Gets the Star5Pointed predefined shape.
	Star6Pointed	Gets the Star6Pointed predefined shape.
	Star7Pointed	Gets the Star7Pointed predefined shape.
	Start	Gets the Start predefined shape.
	Tape	Gets the Tape predefined shape.

	Terminator	Gets the Terminator predefined shape.
	TransmittalTape	Gets the TransmittalTape predefined shape.
	Trapezoid	Gets the Trapezoid predefined shape.

See Also

Shapes Class
MindFusion.Diagramming Namespace

Shapes Properties

Shapes.Actor Property

Gets the Actor predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Actor { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Actor As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Alternative Property

Gets the Alternative predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Alternative { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Alternative As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class

Shapes.Arrow1 Property

Gets the Arrow1 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Arrow1 { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Arrow1 As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace
Table of Predefined Shapes

Shapes.Arrow2 Property

Gets the Arrow2 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Arrow2 { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Arrow2 As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow3 Property

Gets the Arrow3 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Arrow3 { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Arrow3 As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow4 Property

Gets the Arrow4 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Arrow4 { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Arrow4 As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow5 Property

Gets the Arrow5 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Arrow5 { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Arrow5 As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow6 Property

Gets the Arrow6 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Arrow6 { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Arrow6 As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow7 Property

Gets the Arrow7 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Arrow7 { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Arrow7 As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow8 Property

Gets the Arrow8 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Arrow8 { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Arrow8 As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Arrow9 Property

Gets the Arrow9 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Arrow9 { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Arrow9 As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BeginLoop Property

Gets the BeginLoop predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BeginLoop { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BeginLoop As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnComplex Property

Gets the BpmnComplex predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnComplex { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnComplex As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnDataBasedXor Property

Gets the BpmnDataBasedXor predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnDataBasedXor { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnDataBasedXor As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndCancel Property

Gets the BpmnEndCancel predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnEndCancel { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnEndCancel As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndCompensation Property

Gets the BpmnEndCompensation predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnEndCompensation { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnEndCompensation As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndError Property

Gets the BpmnEndError predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnEndError { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnEndError As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndLink Property

Gets the BpmnEndLink predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnEndLink { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnEndLink As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndMessage Property

Gets the BpmnEndMessage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnEndMessage { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnEndMessage As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndMultiple Property

Gets the BpmnEndMultiple predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnEndMultiple { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnEndMultiple As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEndTerminate Property

Gets the BpmnEndTerminate predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnEndTerminate { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnEndTerminate As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnEventBasedXor Property

Gets the BpmnEventBasedXor predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnEventBasedXor { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnEventBasedXor As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnInclusive Property

Gets the BpmnInclusive predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnInclusive { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnInclusive As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateCancel Property

Gets the BpmnIntermediateCancel predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnIntermediateCancel { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnIntermediateCancel As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateCompensation Property

Gets the BpmnIntermediateCompensation predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static Shape BpmnIntermediateCompensation { get; }
```

Visual Basic

```
Public Shared ReadOnly Property BpmnIntermediateCompensation As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateError Property

Gets the BpmnIntermediateError predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static Shape BpmnIntermediateError { get; }
```

Visual Basic

```
Public Shared ReadOnly Property BpmnIntermediateError As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateLink Property

Gets the BpmnIntermediateLink predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static Shape BpmnIntermediateLink { get; }
```

Visual Basic

```
Public Shared ReadOnly Property BpmnIntermediateLink As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateMessage Property

Gets the BpmnIntermediateMessage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnIntermediateMessage { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnIntermediateMessage As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateMultiple Property

Gets the BpmnIntermediateMultiple predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnIntermediateMultiple { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnIntermediateMultiple As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateRule Property

Gets the BpmnIntermediateRule predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnIntermediateRule { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnIntermediateRule As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnIntermediateTimer Property

Gets the BpmnIntermediateTimer predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnIntermediateTimer { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnIntermediateTimer As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnParallelFork Property

Gets the BpmnParallelFork predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnParallelFork { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnParallelFork As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnStartLink Property

Gets the BpmnStartLink predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnStartLink { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnStartLink As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnStartMessage Property

Gets the BpmnStartMessage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnStartMessage { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnStartMessage As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnStartMultiple Property

Gets the BpmnStartMultiple predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnStartMultiple { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnStartMultiple As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnStartRule Property

Gets the BpmnStartRule predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnStartRule { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnStartRule As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.BpmnStartTimer Property

Gets the BpmnStartTimer predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape BpmnStartTimer { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property BpmnStartTimer As Shape
```

Property Value

See Also

Shapes.Cloud Property

Gets the Cloud predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Cloud { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Cloud As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace
Table of Predefined Shapes

Shapes.Collate Property

Gets the Collate predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Collate { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Collate As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ConeDown Property

Gets the ConeDown predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape ConeDown { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property ConeDown As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ConeUp Property

Gets the ConeUp predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape ConeUp { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property ConeUp As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ConnectedIssues Property

Gets the ConnectedIssues predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape ConnectedIssues { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property ConnectedIssues As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.CreateRequest Property

Gets the CreateRequest predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape CreateRequest { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property CreateRequest As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Cross Property

Gets the Cross predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Cross { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Cross As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Cube Property

Gets the Cube predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Cube { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Cube As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Cylinder Property

Gets the Cylinder predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Cylinder { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Cylinder As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Database Property

Gets the Database predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Database { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Database As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DataTransmition Property

Gets the DataTransmition predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape DataTransmition { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property DataTransmition As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DDelay Property

Gets the DDelay predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape DDelay { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property DDelay As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Decagon Property

Gets the Decagon predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Decagon { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Decagon As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Decision Property

Gets the Decision predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Decision { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Decision As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Decision2 Property

Gets the Decision2 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Decision2 { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Decision2 As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Delay Property

Gets the Delay predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Delay { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Delay As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DInDelay Property

Gets the DInDelay predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape DInDelay { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property DInDelay As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DirectAccessStorage Property

Gets the DirectAccessStorage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape DirectAccessStorage { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property DirectAccessStorage As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DiskStorage Property

Gets the DiskStorage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape DiskStorage { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property DiskStorage As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Display Property

Gets the Display predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Display { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Display As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DividedEvent Property

Gets the DividedEvent predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape DividedEvent { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property DividedEvent As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DividedProcess Property

Gets the DividedProcess predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape DividedProcess { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property DividedProcess As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Document Property

Gets the Document predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Document { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Document As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Donut Property

Gets the Donut predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Donut { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Donut As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.DOutDelay Property

Gets the DOutDelay predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape DOutDelay { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property DOutDelay As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Ellipse Property

Gets the Ellipse predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Ellipse { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Ellipse As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.EndLoop Property

Gets the EndLoop predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape EndLoop { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property EndLoop As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ExternalOrganization Property

Gets the ExternalOrganization predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape ExternalOrganization { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property ExternalOrganization As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ExternalProcess Property

Gets the ExternalProcess predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape ExternalProcess { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property ExternalProcess As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.File Property

Gets the File predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape File { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property File As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.FramedRectangle Property

Gets the FramedRectangle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape FramedRectangle { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property FramedRectangle As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Gate Property

Gets the Gate predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Gate { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Gate As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Heart Property

Gets the Heart predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Heart { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Heart As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Heptagon Property

Gets the Heptagon predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Heptagon { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Heptagon As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Hourglass Property

Gets the Hourglass predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Hourglass { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Hourglass As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Input Property

Gets the Input predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Input { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Input As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.InternalStorage Property

Gets the InternalStorage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape InternalStorage { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property InternalStorage As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Interrupt Property

Gets the Interrupt predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Interrupt { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Interrupt As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Interrupt2 Property

Gets the Interrupt2 predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Interrupt2 { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Interrupt2 As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.IsoProcess Property

Gets the IsoProcess predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape IsoProcess { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property IsoProcess As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Junction Property

Gets the Junction predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Junction { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Junction As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Lightning Property

Gets the Lightning predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Lightning { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Lightning As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.LinedDocument Property

Gets the LinedDocument predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape LinedDocument { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property LinedDocument As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.ManualOperation Property

Gets the ManualOperation predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape ManualOperation { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property ManualOperation As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Merge Property

Gets the Merge predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Merge { get; }
```

Visual Basic

```
Public Shared ReadOnly Property Merge As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MessageFromUser Property

Gets the MessageFromUser predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static Shape MessageFromUser { get; }
```

Visual Basic

```
Public Shared ReadOnly Property MessageFromUser As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MessageToUser Property

Gets the MessageToUser predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static Shape MessageToUser { get; }
```

Visual Basic

```
Public Shared ReadOnly Property MessageToUser As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Microform Property

Gets the Microform predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Microform { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Microform As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MicroformProcessing Property

Gets the MicroformProcessing predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape MicroformProcessing { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property MicroformProcessing As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MicroformRecording Property

Gets the MicroformRecording predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape MicroformRecording { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property MicroformRecording As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MultiDocument Property

Gets the MultiDocument predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape MultiDocument { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property MultiDocument As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.MultiProc Property

Gets the MultiProc predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape MultiProc { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property MultiProc As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Octagon Property

Gets the Octagon predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Octagon { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Octagon As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.OfflineStorage Property

Gets the OfflineStorage predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape OfflineStorage { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property OfflineStorage As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.OffpageConnection Property

Gets the OffpageConnection predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape OffpageConnection { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property OffpageConnection As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.OffpageReference Property

Gets the OffpageReference predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape OffpageReference { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property OffpageReference As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Or Property

Gets the Or predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Or { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Or As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Pentagon Property

Gets the Pentagon predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Pentagon { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Pentagon As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Plaque Property

Gets the Plaque predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Plaque { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Plaque As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PrimitiveFromCall Property

Gets the PrimitiveFromCall predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape PrimitiveFromCall { get; }
```

Visual Basic

```
Public Shared ReadOnly Property PrimitiveFromCall As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PrimitiveToCall Property

Gets the PrimitiveToCall predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static Shape PrimitiveToCall { get; }
```

Visual Basic

```
Public Shared ReadOnly Property PrimitiveToCall As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Procedure Property

Gets the Procedure predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static Shape Procedure { get; }
```

Visual Basic

```
Public Shared ReadOnly Property Procedure As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Process Property

Gets the Process predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Process { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Process As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Prysm Property

Gets the Prysm predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Prysm { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Prysm As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PunchedCard Property

Gets the PunchedCard predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape PunchedCard { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property PunchedCard As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PyramidDown Property

Gets the PyramidDown predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape PyramidDown { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property PyramidDown As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.PyramidUp Property

Gets the PyramidUp predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape PyramidUp { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property PyramidUp As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Rectangle Property

Gets the Rectangle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Rectangle { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Rectangle As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.RightTriangle Property

Gets the RightTriangle predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape RightTriangle { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property RightTriangle As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.RoundRect Property

Gets the RoundRect predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape RoundRect { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property RoundRect As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.RSave Property

Gets the RSave predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape RSave { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property RSave As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Save Property

Gets the Save predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Save { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Save As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Sort Property

Gets the Sort predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Sort { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Sort As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Sphere Property

Gets the Sphere predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Sphere { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Sphere As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star Property

Gets the Star predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Star { get; }
```

Visual Basic

```
Public Shared ReadOnly Property Star As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star16Pointed Property

Gets the Star16Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static Shape Star16Pointed { get; }
```

Visual Basic

```
Public Shared ReadOnly Property Star16Pointed As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star24Pointed Property

Gets the Star24Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static Shape Star24Pointed { get; }
```

Visual Basic

```
Public Shared ReadOnly Property Star24Pointed As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star32Pointed Property

Gets the Star32Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Star32Pointed { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Star32Pointed As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star4Pointed Property

Gets the Star4Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Star4Pointed { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Star4Pointed As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star5Pointed Property

Gets the Star5Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Star5Pointed { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Star5Pointed As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star6Pointed Property

Gets the Star6Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Star6Pointed { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Star6Pointed As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Star7Pointed Property

Gets the Star7Pointed predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Star7Pointed { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Star7Pointed As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Start Property

Gets the Start predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Start { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Start As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Tape Property

Gets the Tape predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Tape { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Tape As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Terminator Property

Gets the Terminator predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Terminator { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Terminator As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.TransmittalTape Property

Gets the TransmittalTape predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape TransmittalTape { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property TransmittalTape As Shape
```

Property Value

An instance of the Shape class representing the predefined shape.

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

Shapes.Trapezoid Property

Gets the Trapezoid predefined shape.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Shape Trapezoid { get; }
```

Visual Basic Copy Code

```
Public Shared ReadOnly Property Trapezoid As Shape
```

Property Value

See Also

Shapes Members
Shapes Class
MindFusion.Diagramming Namespace

ShowAnchors Enumeration

Specifies when anchors points should be displayed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum ShowAnchors
```

Visual Basic Copy Code

```
Public Enum ShowAnchors
```

Members

	Member name	Description
	Always	Anchor point marks are always painted.
	Never	Anchor point marks are never painted.
	Auto	Anchor point marks are shown for the node under the mouse cursor.
	Selected	Anchor point marks are painted for the selected node.

Remarks

Members of this enumeration are used as values of the ShowAnchors property of the Diagram class. **Auto** and **Selected** can be combined using the bitwise or operator.

See Also

MindFusion.Diagramming Namespace
ShowAnchors Property
MarkStyle Enumeration

SimpleShape Enumeration

Defines the possible values for Shape property of tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum SimpleShape
```

Visual Basic Copy Code

```
Public Enum SimpleShape
```

Members

	Member name	Description
	Rectangle	Sets table's shape to rectangle.
	RoundedRectangle	Sets table's shape to rounded rectangle.

See Also

MindFusion.Diagramming Namespace

SmoothShadowEffect Class

Represents an effect that smooths the shadow of nodes. This effect does nothing if shadows are not visible.

Syntax

C# Copy Code

```
public sealed class SmoothShadowEffect : NodeEffect
```

Visual Basic Copy Code

```
Public NotInheritable Class SmoothShadowEffect
    Inherits NodeEffect
```

Remarks

This effect can be applied to the nodes in a Diagram by adding an instance of this class to the NodeEffects collection. In addition, this effect will be active only when the Diagram.EffectsProvider property is set to an object implementing the GenerateShadow method of the IEffectsProvider interface.

The following image illustrates the Shadow effect applied to a rectangular node:



Example

The following example demonstrates how to apply smooth shadow effect to the nodes of an existing diagram.

C# Copy Code

```
diagram.NodeEffects.Add(new SmoothShadowEffect());
```

Visual Basic Copy Code

```
diagram.NodeEffects.Add(New SmoothShadowEffect())
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.NodeEffect
    MindFusion.Diagramming.SmoothShadowEffect

See Also

SmoothShadowEffect Members
MindFusion.Diagramming Namespace

SmoothShadowEffect Members

Represents an effect that smooths the shadow of nodes. This effect does nothing if shadows are not visible.

The following tables list the members exposed by the SmoothShadowEffect type.

Public Constructors

	Name	Description
	SmoothShadowEffect	Initializes a new instance of the **SmoothShadowEffect** class.

Public Properties

	Name	Description
	Dispersion	Gets or sets the shadow dispersion.

Public Methods

	Name	Description
	Clone	NodeEffect.Clone override.

Public Events

	Name	Description
	PropertyChanged	INotifyPropertyChanged.PropertyChanged implementation. (Inherited from NodeEffect.)

See Also

SmoothShadowEffect Class
MindFusion.Diagramming Namespace

SmoothShadowEffect Constructor

Initializes a new instance of the SmoothShadowEffect class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SmoothShadowEffect ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SmoothShadowEffect Class
MindFusion.Diagramming Namespace

SmoothShadowEffect Methods

SmoothShadowEffect.Clone Method

NodeEffect.Clone override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Object Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Object
```

Return Value

See Also

SmoothShadowEffect Members
SmoothShadowEffect Class
MindFusion.Diagramming Namespace

SmoothShadowEffect Properties

SmoothShadowEffect.Dispersion Property

Gets or sets the shadow dispersion.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Dispersion { get; set; }
```

Visual Basic Copy Code

```
Public Property Dispersion As Integer
```

Property Value

Remarks

This property is hidden because it slows down performance with bigger values.

See Also

SmoothShadowEffect Members
SmoothShadowEffect Class
MindFusion.Diagramming Namespace

SmoothShadowEffect Events

SnapToAnchor Enumeration

Specifies when link ends are snapped to nearest anchor points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum SnapToAnchor
```

Visual Basic	Copy Code
`Public Enum SnapToAnchor`	

Members

	Member name	Description
	OnCreate	Snap to anchor point when creating a new link.
	OnCreateOrModify	Snap to anchor point when creating or modifying a link.

Remarks

Members of this enumeration are used as values of the SnapToAnchor property of the Diagram class.

See Also

MindFusion.Diagramming Namespace
SnapToAnchor Property
ShowAnchors Enumeration

Style Class

Represents an unordered dynamic collection of properties that can be identified by name.

Syntax

C#	Copy Code
`public class Style`	

Visual Basic	Copy Code
`Public Class Style`	

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle
    MindFusion.Diagramming.DiagramStyle

See Also

Style Members
MindFusion.Diagramming Namespace

Style Members

Represents an unordered dynamic collection of properties that can be identified by name.

The following tables list the members exposed by the Style type.

Public Constructors

	Name	Description
	Style	Initializes a new instance of the **Style** class.

Public Methods

	Name	Description

	GetProperties	Gets all properties in the style.
	GetProperty	Returns the property with the specified name, if any.
S	RegisterType	Associates a type with an alias.

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property.
	IsPropertySet	Checks whether the specified property has a value.
	LoadFromXml	Loads the style content from an XML element.
	OnPropertyChanged	Raises the PropertyChanged event.
	RegisterProperty	Registers a new property with the specified name and type.
	SaveToXml	Saves the style content into an XML element.
	SetValue	Sets the value of the specified property.
	UnsetValue	Clears the value of the specified property.

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed.

See Also

Style Class
MindFusion.Diagramming Namespace

Style Constructor

Initializes a new instance of the Style class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Style ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

Style Class
MindFusion.Diagramming Namespace

Style Methods

Style.GetProperties Method

Gets all properties in the style.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IEnumerable<IProperty> GetProperties ()
```

Visual Basic Copy Code

```
Public Function GetProperties () As IEnumerable(Of IProperty)
```

Return Value

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.GetProperty Method

Returns the property with the specified name, if any.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IProperty GetProperty (
    string name
)
```

Visual Basic Copy Code

```
Public Function GetProperty( _
    name As String _
) As IProperty
```

Parameters

name

Return Value

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.GetValue Method

Gets the value of the specified property.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected Object GetValue (
    Object property
)
```

Visual Basic Copy Code

```
Protected Function GetValue( _
    property As Object _
) As Object
```

Parameters

property
The property to get.

Return Value

The value of the property or **null** (**Nothing** in Visual Basic) if the property is not defined.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.IsPropertySet Method

Checks whether the specified property has a value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected bool IsPropertySet (
    Object property
)
```

Visual Basic Copy Code

```
Protected Function IsPropertySet( _
    property As Object _
) As Boolean
```

Parameters

property
The property to check.

Return Value

true if the property has a local value; otherwise, **false**.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.LoadFromXml Method

Loads the style content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** containing the style's data.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.OnPropertyChanged Method

Raises the PropertyChanged event.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void OnPropertyChanged (
    StylePropertyChangedEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnPropertyChanged( _
    e As StylePropertyChangedEventArgs _
)
```

Parameters

e

An object providing the event data.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.RegisterProperty Method

Registers a new property with the specified name and type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected Object RegisterProperty (
    string name
)
```

Visual Basic Copy Code

```
Protected Function RegisterProperty( _
    name As String _
) As Object
```

Parameters

name
The name of the property.

Return Value

An object that can be used to identify the property in subsequent calls to methods like SetValue and GetValue.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.RegisterType Method

Associates a type with an alias.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static void RegisterType (
    Type type,
    string alias
)
```

Visual Basic Copy Code

```
Public Shared Sub RegisterType( _
    type As Type, _
    alias As String _
)
```

Parameters

type
The type to associate.

alias
The unique alias for the type.

Remarks

The alias is used when saving and loading objects of the specified type during style serialization and deserialization.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.SaveToXml Method

Saves the style content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected virtual void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** the style's data should be stored into.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.SetValue Method

Sets the value of the specified property.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SetValue (
    Object property,
    Object value
)
```

Visual Basic Copy Code

```
Protected Sub SetValue( _
    property As Object, _
    value As Object _
)
```

Parameters

property
The property to set.

value
The new property value.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style.UnsetValue Method

Clears the value of the specified property.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void UnsetValue (
    Object property
)
```

Visual Basic Copy Code

```
Protected Sub UnsetValue( _
    property As Object _
)
```

Parameters

property
The property to clear.

See Also

Style Members
Style Class
MindFusion.Diagramming Namespace

Style Events

StylePropertyChangedEventArgs Class

Contains the arguments passed to handlers of the PropertyChanged event.

Syntax

C# Copy Code

```
public class StylePropertyChangedEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class StylePropertyChangedEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
System.EventArgs
MindFusion.Diagramming.StylePropertyChangedEventArgs

See Also

StylePropertyChangedEventArgs Members
MindFusion.Diagramming Namespace

StylePropertyChangedEventArgs Members

Contains the arguments passed to handlers of the PropertyChanged event.

The following tables list the members exposed by the StylePropertyChangedEventArgs type.

Public Properties

	Name	Description
	NewValue	Gets the new (current) value of the changed property.
	OldValue	Gets the previous value of the changed property.
	Property	Gets the property that has changed.

See Also

StylePropertyChangedEventArgs Class
MindFusion.Diagramming Namespace

StylePropertyChangedEventArgs Properties

StylePropertyChangedEventArgs.NewValue Property

Gets the new (current) value of the changed property.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object NewValue { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NewValue As Object
```

Property Value

An object representing the new property value.

See Also

StylePropertyChangedEventArgs Members
StylePropertyChangedEventArgs Class
MindFusion.Diagramming Namespace

StylePropertyChangedEventArgs.OldValue Property

Gets the previous value of the changed property.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object OldValue { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property OldValue As Object
```

Property Value

An object representing the old property value.

See Also

StylePropertyChangedEventArgs Members
StylePropertyChangedEventArgs Class
MindFusion.Diagramming Namespace

StylePropertyChangedEventArgs.Property Property

Gets the property that has changed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Property { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Property As Object
```

Property Value

An object identifying the changed property.

See Also

StylePropertyChangedEventArgs Members
StylePropertyChangedEventArgs Class
MindFusion.Diagramming Namespace

SvgNode Class

Represents nodes that can render SVG drawings.

Syntax

C# Copy Code

```
public class SvgNode : ShapeNode
```

Visual Basic Copy Code

```
Public Class SvgNode
    Inherits ShapeNode
```

Remarks

An **SvgNode** instance represents nodes that can display SVG drawings in addition to the rendering provided by the base ShapeNode class. The SVG drawing is rendered after the associated Image and just before the DrawNode event is raised with CustomDraw.Additional.

The SVG drawing of the node is specified through its Content property. The SVG graphics is loaded and associated with the node through an **SvgContent** object, which should be initialized and then assigned to this property.

The following code illustrates how to load an SVG graphics from a local file and associate it with an existing **SvgNode** (referenced by the *svgNode* variable):

C# Copy Code

```
SvgContent content = new SvgContent();
content.Parse(@"c:\image.svg");
svgNode.Content = content;
```

VB.NET Copy Code

```
Dim content As New SvgContent()
content.Parse("c:\image.svg")
svgNode.Content = content
```

Note

The **SvgContent** class is defined in the *MindFusion.Svg.dll* assembly. You need to add a reference to this assembly in the target project and import the necessary namespaces.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.ShapeNode
        MindFusion.Diagramming.SvgNode

See Also

SvgNode Members
MindFusion.Diagramming Namespace

SvgNode Members

Represents nodes that can render SVG drawings.

The following tables list the members exposed by the SvgNode type.

Public Constructors

	Name	Description
	SvgNode	Overloaded. Initializes a new instance of the **SvgNode** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowFlip	Gets or sets a value indicating whether the node's shape flips when an edge is dragged over its opposite edge. (Inherited from ShapeNode.)

	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	Content	Gets or set an **SvgContent** instance representing the SVG drawing rendered in this node.
	ContentAlign	Gets or sets the alignment of the SvgContent displayed in this node.
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	CustomDraw	Gets or sets the type of custom drawing performed on this shape node. (Inherited from ShapeNode.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled. (Inherited from ShapeNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	FlipX	Gets or sets a value indicating whether the node's shape is flipped horizontally. (Inherited from ShapeNode.)
	FlipY	Gets or sets a value indicating whether the node's shape is flipped vertically. (Inherited from ShapeNode.)
	Font	Gets or sets the font used to render the node's text. (Inherited from ShapeNode.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)

	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	Image	Gets or sets the image that is displayed in this shape node. (Inherited from ShapeNode.)
	ImageAlign	Gets or sets the alignment of the image displayed in this shape node. (Inherited from ShapeNode.)
	ImagePadding	Gets or sets the padding distance between node's borders and **Image**. (Inherited from ShapeNode.)
	ImageUrl	Gets or sets a URL used to load an image into this node. (Inherited from ShapeNode.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	PlainText	Gets the plain text of this shape node. (Inherited from ShapeNode.)
	PolygonalTextLayout	Gets or sets a value indicating whether the node's Text is formatted inside non-rectangular outline. (Inherited from ShapeNode.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotateImage	Gets or sets a value indicating whether the **Image** should be rotated when the shape node is rotated. (Inherited from ShapeNode.)
	RotateText	Gets or sets a value indicating whether the **Text** should be rotated when the shape node is rotated. (Inherited from ShapeNode.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)

	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Shape	Gets or sets a reference to the node's geometric shape definition. (Inherited from ShapeNode.)
	ShapeOrientation	Gets or sets the angle specifying the orientation with which a shape template is applied to the node bounds. (Inherited from ShapeNode.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	SvgUrl	Gets or sets a URL used to load an SVG file into this node.
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed inside the shape node. (Inherited from ShapeNode.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of the text that is displayed inside shape nodes. (Inherited from ShapeNode.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from ShapeNode.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Transparent	Gets or sets a value indicating whether this shape node is transparent. (Inherited from ShapeNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description

	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from ShapeNode.)
	CreateConnectionPoint	DiagramNode.CreateConnectionPoint override. (Inherited from ShapeNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramNode.DrawHandles override. Renders the adjustment handles of this item. (Inherited from ShapeNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Draws the node on the specified IGraphics surface. (Inherited from ShapeNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified **IGraphics** surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Draws the shadow of this node on the specified **IGraphics** surface. (Inherited from ShapeNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)

	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from ShapeNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	Node.GetOutlinePoly override. (Inherited from ShapeNode.)
	GetRepaintRect	DiagramNode.GetRepaintRect override. (Inherited from ShapeNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramNode.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from ShapeNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	DiagramItem.OnLoad override. (Inherited from ShapeNode.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	ResizeToFitImage	Makes the shape node as big as the image it displays. (Inherited from ShapeNode.)
	ResizeToFitText	Makes the shape node big enough to display its text without clipping. (Inherited from ShapeNode.)

	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramNode.AllowModify override. Invoked during interactive modification of items to test whether the operation can be completed in the current context. (Inherited from ShapeNode.)
	CancelModify	DiagramNode.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from ShapeNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramNode.CompleteModify override. Invoked when the item modification has been completed. (Inherited from ShapeNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override.
	CreateState	DiagramNode.CreateState override; creates a ShapeNodeState instance. (Inherited from ShapeNode.)
	Dispose	Disposes any disposable resources managed by this shape. (Inherited from ShapeNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)

	GetImageRect	Overloaded. Gets the destination rectangle of the node's image. (Inherited from ShapeNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a GraphicsPath object for this node as though it is contained in the specified rectangle. (Inherited from ShapeNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object.
	LoadFromXml	Loads the item content from an XML element.
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	Called when the item is added to a **Diagram**, either by users, programmatically or via undo / redo. (Inherited from ShapeNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	Called when the Diagram's MeasureUnit changes. (Inherited from ShapeNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	DiagramNode.OnRotationAngleChanged override. (Inherited from ShapeNode.)
	OnStyleChanged	DiagramItem.OnStyleChanged override. (Inherited from ShapeNode.)
	OnTextChanged	DiagramItem.OnTextChanged override. (Inherited from ShapeNode.)
	OnTextFormatChanged	DiagramItem.OnTextFormatChanged override. (Inherited from ShapeNode.)
	OnUpdateBounds	DiagramNode.OnUpdateBounds override. (Inherited from ShapeNode.)
	OnZoomChanged	Called when the Diagram's **ZoomFactor** changes. (Inherited from ShapeNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramNode.RestoreProperties override.
	RestoreState	DiagramNode.RestoreState override. Restores the state of this diagram item. (Inherited from ShapeNode.)
	SaveProperties	DiagramNode.SaveProperties override.
	SaveState	DiagramNode.SaveState override. Saves the state of this diagram item. (Inherited from ShapeNode.)
	SaveToJson	Saves the node as a JSON object.
	SaveToXml	Saves the item content into an XML element.

	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramNode.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from ShapeNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	**DiagramNode.UpdateModify** override. Invoked during interactive modification of this item. (Inherited from ShapeNode.)

See Also

SvgNode Class
MindFusion.Diagramming Namespace

SvgNode Constructor

Initializes a new instance of the SvgNode class.

Overload List

Name	Description
SvgNode ()	Initializes a new instance of the **SvgNode** class.
SvgNode (Diagram)	Initializes a new instance of the **SvgNode** class with default values supplied form the specified Diagram.
SvgNode (SvgNode)	Initializes a new instance of the **SvgNode** class using the specified node as a prototype.

See Also

SvgNode Class
MindFusion.Diagramming Namespace

SvgNode Constructor ()

Initializes a new instance of the SvgNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNode ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SvgNode Constructor Overload List
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode Constructor (Diagram)

Initializes a new instance of the SvgNode class with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNode (
    Diagram parent
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram _
)
```

Parameters

parent
The Diagram from which to obtain default values for the node properties.

See Also

SvgNode Constructor Overload List
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode Constructor (SvgNode)

Initializes a new instance of the SvgNode class using the specified node as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNode (
    SvgNode prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As SvgNode _
)
```

Parameters

prototype
The node to use as a prototype. The properties of the new node are initialized from the property values of this node.

See Also

SvgNode Constructor Overload List
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode Methods

SvgNode.CreateProperties Method

DiagramItem.CreateProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemProperties CreateProperties ()
```

Visual Basic Copy Code

```
Protected Overrides Function CreateProperties () As DiagramItemProperties
```

Return Value

A DiagramItemProperties-derived object containing a snapshot of the item's properties.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.LoadFromJson Method

Loads the node from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the node's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.LoadFromXml Method

Loads the item content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** containing the item's data.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.RestoreProperties Method

DiagramNode.RestoreProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties-derived object containing a snapshot of the item's properties.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.SaveProperties Method

DiagramNode.SaveProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties-derived object where the item's properties should be saved.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.SaveToJson Method

Saves the node as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject instance to contain the serialized data.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.SaveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** the item's data should be stored into.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode Properties

SvgNode.Content Property

Gets or set an **SvgContent** instance representing the SVG drawing rendered in this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgContent Content { get; set; }
```

Visual Basic Copy Code

```
Public Property Content As SvgContent
```

Property Value

An instance of the **MindFusion.Svg.SvgContent** class representing the SVG drawing.

Remarks

The **SvgContent** class is defined in the *MindFusion.Svg.dll* file. You must add a reference to this assembly to the target project.

Example

The following example illustrates how to create an **SvgContent** object from a local SVG file and associate it with an existing SvgNode object. The **SvgNode** object is referenced by the *node* variable.

C# Copy Code

```
SvgContent content = new SvgContent();
content.Parse(@"c:\image.svg");
node.Content = content;
```

Visual Basic Copy Code

```
Dim content As New SvgContent()
content.Parse("c:\image.svg")
node.Content = content
```

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.ContentAlign Property

Gets or sets the alignment of the SvgContent displayed in this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ImageAlign ContentAlign { get; set; }
```

Visual Basic Copy Code

```
Public Property ContentAlign As ImageAlign
```

Property Value

A member of the ImageAlign enumeration. The default is **Fit**.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNode.SvgUrl Property

Gets or sets a URL used to load an SVG file into this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string SvgUrl { get; set; }
```

Visual Basic Copy Code

```
Public Property SvgUrl As String
```

Property Value

A string specifying the SVG file location.

Remarks

At this time this property is used only in web versions of the component (ASP.NET and ASP.NET MVC). Note that the MIME type of .svg files on server should be set to image/svg+xml.

See Also

SvgNode Members
SvgNode Class
MindFusion.Diagramming Namespace

SvgNodeProperties Class

Stores the values of SvgNode properties during undo / redo operations. If you need to implement undo support for the additional properties defined by a class derived from **SvgNode**, do so by deriving from **SvgNodeProperties** and overriding the SvgNode.SaveProperties, SvgNode.RestoreProperties, and SvgNode.CreateProperties methods.

Syntax

C# Copy Code

```
public class SvgNodeProperties : ShapeNodeProperties
```

Visual Basic Copy Code

```
Public Class SvgNodeProperties
    Inherits ShapeNodeProperties
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItemProperties
    MindFusion.Diagramming.DiagramNodeProperties
      MindFusion.Diagramming.ShapeNodeProperties
        MindFusion.Diagramming.SvgNodeProperties

See Also

SvgNodeProperties Members
MindFusion.Diagramming Namespace

SvgNodeProperties Members

Stores the values of SvgNode properties during undo / redo operations. If you need to implement undo support for the additional properties defined by a class derived from **SvgNode**, do so by deriving from SvgNodeProperties and overriding the SvgNode.SaveProperties, SvgNode.RestoreProperties, and SvgNode.CreateProperties methods.

The following tables list the members exposed by the **SvgNodeProperties** type.

Public Constructors

	Name	Description
	SvgNodeProperties	Initializes a new instance of the **SvgNodeProperties** class.

Public Fields

	Name	Description
	AllowFlip	For internal use. (Inherited from ShapeNodeProperties.)
	FlipX	For internal use. (Inherited from ShapeNodeProperties.)
	FlipY	For internal use. (Inherited from ShapeNodeProperties.)
	ItemClass	For internal use. (Inherited from DiagramItemProperties.)
	labels	For internal use. (Inherited from DiagramNodeProperties.)
	RotateText	For internal use. (Inherited from ShapeNodeProperties.)
	Selected	For internal use. (Inherited from DiagramItemProperties.)
	Shape	For internal use. (Inherited from ShapeNodeProperties.)
	ShapeOrientation	For internal use. (Inherited from ShapeNodeProperties.)
	Text	For internal use. (Inherited from ShapeNodeProperties.)
	TextPadding	For internal use. (Inherited from DiagramItemProperties.)
	Transparent	For internal use. (Inherited from ShapeNodeProperties.)
	UsePolyTextLt	For internal use. (Inherited from ShapeNodeProperties.)
	ZIndex	For internal use. (Inherited from DiagramItemProperties.)

Public Methods

	Name	Description
	FreeResources	DiagramItemProperties.FreeResources override. (Inherited from ShapeNodeProperties.)

See Also

SvgNodeProperties Class
MindFusion.Diagramming Namespace

SvgNodeProperties Constructor

Initializes a new instance of the SvgNodeProperties class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNodeProperties ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SvgNodeProperties Class
MindFusion.Diagramming Namespace

SvgNodeStyle Class

Represents a reusable set of properties that define the appearance of SVG nodes.

Syntax

C# Copy Code

```
public class SvgNodeStyle : DiagramNodeStyle
```

Visual Basic Copy Code

```
Public Class SvgNodeStyle
    Inherits DiagramNodeStyle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle
      MindFusion.Diagramming.DiagramNodeStyle
        MindFusion.Diagramming.SvgNodeStyle

See Also

SvgNodeStyle Members
MindFusion.Diagramming Namespace

SvgNodeStyle Members

Represents a reusable set of properties that define the appearance of SVG nodes.

The following tables list the members exposed by the SvgNodeStyle type.

Public Constructors

	Name	Description
	SvgNodeStyle	Initializes a new instance of the **SvgNodeStyle** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of the item. (Inherited from DiagramItemStyle.)

	Effects	Gets a list of the effects applied to the node. (Inherited from DiagramNodeStyle.)
	FontFamily	Gets or sets the font family of the item text. (Inherited from DiagramItemStyle.)
	FontSize	Gets or sets the font size of the item text. (Inherited from DiagramItemStyle.)
	FontStyle	Gets or sets the font style of the item text. (Inherited from DiagramItemStyle.)
	FontUnit	Gets or sets the font unit of the item text. (Inherited from DiagramItemStyle.)
	ShadowBrush	Gets or sets the brush used to paint the shadow of the item. (Inherited from DiagramItemStyle.)
	Stroke	Gets or sets the brush used to paint the outline of the item. (Inherited from DiagramItemStyle.)
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items. (Inherited from DiagramItemStyle.)
	StrokeThickness	Gets or sets the thickness of the item's outline, expressed in the current units of measure. (Inherited from DiagramItemStyle.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItemStyle.)

Public Methods

	Name	Description
	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
S	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JsonObject. (Inherited from DiagramNodeStyle.)
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)
	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a JsonObject. (Inherited from DiagramNodeStyle.)
	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description

	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

SvgNodeStyle Class
MindFusion.Diagramming Namespace

SvgNodeStyle Constructor

Initializes a new instance of the SvgNodeStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgNodeStyle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SvgNodeStyle Class
MindFusion.Diagramming Namespace

SvgNodeStyle Events

TableConnectionStyle Enumeration

Specifies the possible values for the ConnectionStyle property of tables.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum TableConnectionStyle
```

Visual Basic Copy Code

```
Public Enum TableConnectionStyle
```

Members

	Member name	Description
	Table	Users are allowed to create relations to a table as whole entity.
	Rows	Users are allowed to create relations to the rows of a table.
	Both	Users are allowed to create relations both to the rows of a table or the the table as a whole.

Remarks

The enumeration provides possible values for the ConnectionStyle property, which specifies whether relation ends are the rows of a table or the table as a whole entity.

See Also

MindFusion.Diagramming Namespace

TableNode Class

Tables are diagram nodes that can contain multiple cells and can be related to other nodes through their rows.

Syntax

C# Copy Code

```
public class TableNode : DiagramNode, Scrollable
```

Visual Basic Copy Code

```
Public Class TableNode
    Inherits DiagramNode
    Implements Scrollable
```

Remarks

Tables in flow diagrams are represented by instances of the **TableNode** class. Probably the most important property of tables is ConnectionStyle, which specifies whether users are allowed to draw links to the rows of a table, to the table as an integral entity, or both.

Tables consist of Cell instances distributed across rows and columns. Number of rows and columns can be set with the RowCount and ColumnCount properties. The width of the columns is set with ColumnWidth property and the height of the rows with the RowHeight property. Cells can be made to span several rows and columns by setting their RowSpan and ColumnSpan properties to values larger than one.

Rows can be assigned to distinct sections of the table. Each section can be collapsed or expanded, hiding or showing all rows in the section except the header one. To define a section, set a Header row. A section consists of all rows after a header row and spans to the next header. Each header row displays a [±] button that allows expanding or collapsing the section interactively. The button appears either on the left of the first cell of a header row or in a separate column, as set in OffsetHeaderRows. Clicking that button raises the TableSectionCollapsed and TableSectionExpanded events. Sections can be collapsed or expanded programmatically using the Expanded property of their header rows.

The structure of a table can be dynamically changed by adding and deleting rows and columns. This can be done with the AddColumn and AddRow methods as well as with the InsertColumn, InsertRow and RedimTable methods. DeleteRow and DeleteColumn methods delete existing rows and columns.

Table cells contain Text, displayed as specified through TextBrush and TextFormat. A table can also have a caption, whose text is set by the Caption property. The height of the caption and the alignment of the text are specified by CaptionHeight and CaptionFormat properties. To get even richer text formatting, set EnableStyledText to **true** and embed HTML-like formatting tags in the caption and cell texts. The ResizeToFitText method resizes all columns and rows so that each table cell fits its text on a single line without clipping it.

To display a text box that lets users edit the table caption or the text of a cell, call the **BeginEdit** method of **DiagramView**. Call **EndEdit** to exit the in-place editing mode. If the view's **AllowInplaceEdit** property is enabled, users can double-click the table caption bar or a cell in order to edit their text.

The interior of tables is filled using their Brush property. The color of the table and cell borders can be set by using the Pen property. The caption bar is painted using the CaptionBackBrush brush.

In order to set an image as table's background use the Image property. The alignment of the image in relation to the table's rectangle is set through the ImageAlign property. The table can be resized to fit the size of its image by calling the ResizeToFitImage method.

The rectangle, which a table occupies, is accessible with the Bounds property. Rows and columns that build the table are stored and accessed by means of the Rows and Columns collections.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.TableNode

See Also

TableNode Members

Tables are diagram nodes that can contain multiple cells and can be related to other nodes through their rows.

The following tables list the members exposed by the TableNode type.

Public Constructors

	Name	Description
	TableNode	Overloaded. Initializes a new instance of the **TableNode** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AllowResizeColumns	Gets or sets a value indicating whether users are allowed to resize table columns.
	AllowResizeRows	Gets or sets a value indicating whether users are allowed to resize table rows.
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Caption	Gets or sets the text displayed as caption of the table.
	CaptionBackBrush	Gets or sets the brush used to paint the caption bar of this table.
	CaptionBrush	Gets or sets the brush used to paint the table caption.
	CaptionFormat	Gets or sets the alignment and line position of the text used as table caption.
	CaptionHeight	Gets or sets the height of the table's caption bar.
	CaptionPlainText	Gets the plain text of the table's caption.
	CellCustomDraw	Gets or sets the type of custom drawing performed on the table cells.
	CellFrameStyle	Gets or sets the style of the table cells.
	CellTextBrush	Gets or sets the brush used to paint texts inside table cells.
	ColumnCount	Gets or sets the number of columns in the table.
	Columns	Gets a list of all columns in the table.

	ColumnWidth	Gets or sets the default width of the table columns.
	ConnectionStyle	Gets or sets a value indicating whether drawing a link from/to this table, should connect a row of the table, or the whole table as single entity.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	CurrentScrollRow	Gets or sets the index of the first visible row in the table.
	CustomDraw	Gets or sets the type of custom drawing performed on this table.
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveCaptionBackBrush	Gets the effective caption back brush of the table.
	EffectiveCaptionBrush	Gets the effective caption brush of the table.
	EffectiveCellTextBrush	Gets the effective text brush of the table cells.
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled.
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used when rendering the table's texts.
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HighlightedRow	Gets or sets the highlighted row.
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	Image	Gets or sets the image that is displayed in this table.
	ImageAlign	Gets or sets the alignment of the image displayed in this table.
	ImagePadding	Gets or sets the padding distance between node's borders and **Image**.

	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Item	Gets the Cell instance at the specified position.
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	OffsetHeaderRows	Gets or sets a value indicating whether the table cells are offset to the right to accommodate for the expand buttons of Header rows.
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RowAnchorPattern	Gets or sets a default anchor pattern for the rows of this table.
	RowCount	Gets or sets the number of rows in the table.
	RowHeight	Gets or sets the default height of table rows.
	Rows	Gets a list of all rows in the table.
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Scrollable	Gets or sets a value indicating whether users are allowed to scroll table rows.
	ScrollUI	Gets or sets what type of scroll UI to show in **TableNode**s.
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Shape	Gets or sets the shape of the table's outline.

	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text.
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	DiagramNode.HasItems override.
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	DiagramNode.AcceptLinks override.
	AddColumn	Adds new empty column at the right end of the table.
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	AddRelation	Creates a relation with the specified table.
	AddRow	Adds new empty row at the bottom of the table.
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	BreakRelation	Breaks the relation with the specified table.
	CellFromPoint	Returns the row and column indices of the cell located at the specified point, taking into consideration the cells span.
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)

	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	DiagramNode.CreateConnectionPoint override.
	DeleteColumn	Deletes the column with the specified index within the table.
	DeleteRow	Deletes the row at the specified position within the table.
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	DiagramNode.DrawAnchors override.
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Draws the node on the specified IGraphics surface.
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified **IGraphics** surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Draws the shadow of this node on the specified **IGraphics** surface.
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectiveCaptionBackBrush	Returns a Brush representing the currently effective caption back brush of the table.
	GetEffectiveCaptionBrush	Returns a Brush representing the currently effective caption brush of the table.
	GetEffectiveCellTextBrush	Returns a Brush representing the currently effective text brush of the table cells.
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	DiagramNode.GetIntersection override.
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)

	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetSpannedCellRect	Overloaded. Calculates the bounding rectangle of the specified cell, taking in consideration the cell's row and column spans.
	GetToolTip	DiagramNode.GetToolTip override.
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override.
	InsertColumn	Inserts new empty column at the specified position within the table.
	InsertRow	Inserts new empty row at the specified position within the table.
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	DiagramItem.OnLoad override.
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RedimTable	Sets new number of rows and columns for this table.
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	ResizeToFitImage	Makes a table as big as the image it displays.
	ResizeToFitText	Overloaded.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)

	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	ScrollDown	Scrolls the table rows downwards.
	ScrollUp	Scrolls the table rows upwards.
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override.
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	Dispose	Disposes any disposable resources managed by this table.
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllIncomingLinks	Node.GetAllIncomingLinks override.
	GetAllOutgoingLinks	Node.GetAllOutgoingLinks override.
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)

	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object.
	LoadFromXml	Loads the item content from an XML element.
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	Called when the Diagram's MeasureUnit changes.
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	DiagramItem.OnStyleChanged override.
	OnTextChanged	DiagramItem.OnTextChanged override.
	OnTextFormatChanged	DiagramItem.OnTextFormatChanged override.
	OnUpdateBounds	DiagramNode.OnUpdateBounds override.
	OnZoomChanged	Called when the **DiagramView.ZoomFactor** changes.
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramNode.RestoreProperties override.
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramNode.SaveProperties override.
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object.
	SaveToXml	Saves the item content into an XML element.
	ScaleElements	DiagramNode.ScaleElements override.
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)

	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it.
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

TableNode Class
MindFusion.Diagramming Namespace

TableNode Constructor

Initializes a new instance of the TableNode class.

Overload List

Name	Description
TableNode ()	Initializes a new instance of the **TableNode** class.
TableNode (Diagram)	Initializes a new instance of the **TableNode** class.
TableNode (TableNode)	Initializes a new instance of the **TableNode** class using the specified node as a prototype.

See Also

TableNode Class
MindFusion.Diagramming Namespace

TableNode Constructor ()

Initializes a new instance of the TableNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public TableNode ()
```

Visual Basic

```
Public Sub New ()
```

See Also

TableNode Constructor Overload List
TableNode Class
MindFusion.Diagramming Namespace

TableNode Constructor (Diagram)

Initializes a new instance of the TableNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public TableNode (
    Diagram parent
)
```

Visual Basic

```
Public New ( _
    parent As Diagram _
)
```

Parameters

parent
The Diagram from which to obtain default values for the node properties.

Remarks

This constructor initializes the TableNode properties with the values specified in the respective default-value properties of the Diagram object. For more information see **Initial Values of Item Properties**.

See Also

TableNode Constructor Overload List
TableNode Class
MindFusion.Diagramming Namespace

TableNode Constructor (TableNode)

Initializes a new instance of the TableNode class using the specified node as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public TableNode (
    TableNode prototype
)
```

Visual Basic

```
Public New ( _
    prototype As TableNode _
)
```

Parameters

prototype
The table node to use as a prototype. The properties of the new table are initialized from the property values of *prototype*.

Remarks

This constructor copies the TableNode property values from the specified prototype node.

See Also

TableNode Constructor Overload List
TableNode Class
MindFusion.Diagramming Namespace

TableNode Methods

TableNode.AcceptLinks Method

DiagramNode.AcceptLinks override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override bool AcceptLinks (
    bool outgoing
)
```

Visual Basic

```
Public Overrides Function AcceptLinks( _
    outgoing As Boolean _
) As Boolean
```

Parameters

outgoing
true to check for outgoing links; otherwise, **false**.

Return Value

true if the node accepts connections of the specified type; otherwise, **false**.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.AddColumn Method

Adds new empty column at the right end of the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int AddColumn ()
```

Visual Basic Copy Code

```
Public Function AddColumn () As Integer
```

Return Value

Returns the index of the newly added column within the Columns collection.

Remarks

Call the method to add new columns at the end of the table. The size of the table is automatically updated and the columns are added to the column collection.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
InsertColumn Method
DeleteColumn Method
RedimTable Method
Columns Property

TableNode.AddRelation Method

Creates a relation with the specified table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLink AddRelation (
    int row,
    Relationship rel,
    TableNode tableNode,
    int relRow
)
```

Visual Basic Copy Code

```
Public Function AddRelation( _
    row As Integer, _
    rel As Relationship, _
    tableNode As TableNode, _
    relRow As Integer _
) As DiagramLink
```

Parameters

row
Zero-based index of a row in this table to be related.

rel
The type of the relation (and the direction of the link).

tableNode
The other table in the relation.

relRow
Zero-based index of the row in the other table.

Return Value

A DiagramLink object representing the relation.

Remarks

Call this method to link rows of two tables.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
BreakRelation Method

TableNode.AddRow Method

Adds new empty row at the bottom of the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int AddRow ()
```

Visual Basic Copy Code

```
Public Function AddRow () As Integer
```

Return Value

Returns the index of the newly added row within the Rows collection.

Remarks

Call the method to add new rows at the end of the table. The size of the table is automatically updated and the rows are added to the row collection.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
InsertRow Method
DeleteRow Method
RedimTable Method
Rows Property

TableNode.BreakRelation Method

Breaks the relation with the specified table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void BreakRelation (
    int row,
    Relationship rel,
    TableNode tableNode
)
```

Visual Basic Copy Code

```
Public Sub BreakRelation( _
    row As Integer, _
```

```
    rel As Relationship, _
    tableNode As TableNode _
)
```

Parameters

row
Zero-based index of a row in this table that is related to a row in another.

rel
The type of the relation (and the direction of the link).

tableNode
The other table in the relation.

Remarks

Call this method to break the link between two tables.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
AddRelation Method

TableNode.CellFromPoint Method

Returns the row and column indices of the cell located at the specified point, taking into consideration the cells span.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool CellFromPoint (
    Point point,
    ref int row,
    ref int col
)
```

Visual Basic Copy Code

```
Public Function CellFromPoint( _
    point As Point, _
    ByRef row As Integer, _
    ByRef col As Integer _
) As Boolean
```

Parameters

point
The point to check.

row
Receives the row index if a cell is found at the specified position.

col
Receives the column index if a cell is found at the specified position.

Return Value

true if a cell of this table contains the specified point; otherwise, false.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.CreateConnectionPoint Method

DiagramNode.CreateConnectionPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override ConnectionPoint CreateConnectionPoint (
    DiagramLink link,
    Point point,
    bool incoming
)
```

Visual Basic Copy Code

```
Public Overrides Function CreateConnectionPoint( _
    link As DiagramLink, _
    point As Point, _
    incoming As Boolean _
) As ConnectionPoint
```

Parameters

link
A DiagramLink that should be connected to this node.

point
The point at which the link should be connected.

incoming
true if *link* is an incoming link; otherwise, **false**.

Return Value

A ConnectionPoint instance representing the connection.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.CreateProperties Method

DiagramItem.CreateProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemProperties CreateProperties ()
```

Visual Basic Copy Code

```
Protected Overrides Function CreateProperties () As DiagramItemProperties
```

Return Value

A DiagramItemProperties -derived object containing a snapshot of the item's properties.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.DeleteColumn Method

Deletes the column with the specified index within the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void DeleteColumn (
    int col
)
```

Visual Basic Copy Code

```
Public Sub DeleteColumn( _
    col As Integer _
)
```

Parameters

col
The zero-based index of the column to be deleted.

Remarks

Call the method to delete a column from the table. The size of the table is automatically updated and the columns are removed from the column collection.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
DeleteRow Method
AddColumn Method
InsertColumn Method
RedimTable Method

TableNode.DeleteRow Method

Deletes the row at the specified position within the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void DeleteRow (
    int row
)
```

Visual Basic Copy Code

```
Public Sub DeleteRow( _
    row As Integer _
)
```

Parameters

row
The zero-based index of the row to be deleted.

Remarks

Call the method to delete a row from the table. The size of the table is automatically updated and the rows are removed from the row collection.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
DeleteColumn Method
AddRow Method
InsertRow Method
RedimTable Method

TableNode.Dispose Method

Disposes any disposable resources managed by this table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void Dispose (
    bool disposing
)
```

Visual Basic Copy Code

```
Protected Overrides Sub Dispose( _
    disposing As Boolean _
)
```

Parameters

disposing
Specifies whether the method is called explicitly or from the garbage collector.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.DrawAnchors Method

DiagramNode.DrawAnchors override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawAnchors (
    IGraphics graphics
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawAnchors( _
    graphics As IGraphics _
)
```

Parameters

graphics
The IGraphics surface to draw the anchor points on.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.DrawLocal Method

Draws the node on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawLocal (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawLocal( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
An IGraphics object where the node should be rendered.

options
Provides options for the drawing operation.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.DrawShadowLocal Method

Draws the shadow of this node on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawShadowLocal (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic

```
Public Overrides Sub DrawShadowLocal( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
An IGraphics object where the shadow should be rendered.

options
Provides options for the drawing operation.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.GetAllIncomingLinks Method

Node.GetAllIncomingLinks override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void GetAllIncomingLinks (
    DiagramLinkCollection links
)
```

Visual Basic

```
Protected Overrides Sub GetAllIncomingLinks( _
    links As DiagramLinkCollection _
)
```

Parameters

links

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.GetAllOutgoingLinks Method

Node.GetAllOutgoingLinks override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected override void GetAllOutgoingLinks (
    DiagramLinkCollection links
)
```

Visual Basic

```
Protected Overrides Sub GetAllOutgoingLinks( _
    links As DiagramLinkCollection _
)
```

Parameters

links

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.GetEffectiveCaptionBackBrush Method

Returns a Brush representing the currently effective caption back brush of the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush GetEffectiveCaptionBackBrush (
    Rect bounds,
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Public Function GetEffectiveCaptionBackBrush( _
    bounds As Rect, _
    dx As Double, _
    dy As Double _
) As Brush
```

Parameters

bounds
The bounds of the new brush.

dx
The horizontal translation of the new brush.

dy
The vertical translation of the new brush.

Return Value

A Brush instance.

Remarks

This method returns either the table's CaptionBackBrush if not **null**, or the CaptionBackBrush property of the table's style.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.GetEffectiveCaptionBrush Method

Returns a Brush representing the currently effective caption brush of the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush GetEffectiveCaptionBrush ()
```

Visual Basic Copy Code

```
Public Function GetEffectiveCaptionBrush () As Brush
```

Return Value

A Brush instance.

Remarks

This method returns either the table's CaptionBrush if not **null**, or the TextBrush property of the table's style.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.GetEffectiveCellTextBrush Method

Returns a Brush representing the currently effective text brush of the table cells.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush GetEffectiveCellTextBrush ()
```

Visual Basic Copy Code

```
Public Function GetEffectiveCellTextBrush () As Brush
```

Return Value

A Brush instance.

Remarks

This method returns either the table's CellTextBrush if not **null**, or the CellTextBrush property of the table's style.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.GetIntersection Method

DiagramNode.GetIntersection override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```

```

```
public override Point GetIntersection (
    Point segmentStart,
    Point segmentEnd
)
```

Visual Basic Copy Code

```
Public Overrides Function GetIntersection( _
    segmentStart As Point, _
    segmentEnd As Point _
) As Point
```

Parameters

segmentStart
The first point of the segment.

segmentEnd
The second point of the segment.

Return Value

A **Microsoft.Maui.Graphics.Point** representing the intersection point.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.GetSpannedCellRect Method

Calculates the bounding rectangle of the specified cell, taking in consideration the cell's row and column spans.

Overload List

Name	Description
TableNode.GetSpannedCellRect (Int32, Int32)	Calculates the bounding rectangle of the cell in the specified row and column also taking in consideration the cell row and column spanning.
TableNode.GetSpannedCellRect (TableNode.Cell)	Calculates the bounding rectangle of the specified cell also taking in consideration the cell row and column spanning.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.GetSpannedCellRect Method (Int32, Int32)

Calculates the bounding rectangle of the cell in the specified row and column also taking in consideration the cell row and column spanning.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect GetSpannedCellRect (
    int row,
    int col
)
```

Visual Basic

```
Public Function GetSpannedCellRect( _
    row As Integer, _
    col As Integer _
) As Rect
```

Parameters

row

An integer value specifying the cell's row.

col

An integer value specifying the cell's column.

Return Value

A **Microsoft.Maui.Graphics.Rect** representing the cell's bounding rectangle.

See Also

GetSpannedCellRect Method Overload List
TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.GetSpannedCellRect Method (TableNode.Cell)

Calculates the bounding rectangle of the specified cell also taking in consideration the cell row and column spanning.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Rect GetSpannedCellRect (
    TableNode.Cell cell
)
```

Visual Basic

```
Public Function GetSpannedCellRect( _
    cell As TableNode.Cell _
) As Rect
```

Parameters

cell

A Cell instance.

Return Value

A **Microsoft.Maui.Graphics.Rect** representing the cell's bounding rectangle.

See Also

GetSpannedCellRect Method Overload List
TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.GetToolTip Method

DiagramNode.GetToolTip override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override string GetToolTip (
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function GetToolTip( _
    point As Point _
) As String
```

Parameters

point

The mouse pointer position.

Return Value

A string specifying the tooltip text.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.HitTestManipulators Method

DiagramItem.HitTestManipulators override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Manipulator HitTestManipulators (
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function HitTestManipulators( _
    point As Point _
) As Manipulator
```

Parameters

point

Return Value

An instance of **Manipulator** -derived class, or **null** if there isn't one found at specified location.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.InsertColumn Method

Inserts new empty column at the specified position within the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void InsertColumn (
    int col
)
```

Visual Basic Copy Code

```
Public Sub InsertColumn( _
    col As Integer _
)
```

Parameters

col
The zero-based index of the column to be inserted.

Remarks

Call the method to add new column at a particular position in the table. The size of the table is automatically updated and the columns are added to the column collection.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
AddColumn Method
DeleteColumn Method
RedimTable Method

TableNode.InsertRow Method

Inserts new empty row at the specified position within the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void InsertRow (
    int row
)
```

Visual Basic Copy Code

```
Public Sub InsertRow( _
    row As Integer _
)
```

Parameters

row
The zero-based index of the row to be inserted.

Remarks

Call the method to add new row at a particular position in the table. The size of the table is automatically updated and the rows are added to the row collection.

See Also

TableNode Members
TableNode Class

TableNode.LoadFromJson Method

Loads the node from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the node's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.LoadFromXml Method

Loads the item content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** containing the item's data.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.OnLoad Method

DiagramItem.OnLoad override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void OnLoad ()
```

Visual Basic Copy Code

```
Public Overrides Sub OnLoad ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.OnMeasureUnitChanged Method

Called when the Diagram's MeasureUnit changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnMeasureUnitChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnMeasureUnitChanged ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.OnStyleChanged Method

DiagramItem.OnStyleChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnStyleChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnStyleChanged ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.OnTextChanged Method

DiagramItem.OnTextChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnTextChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnTextChanged ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.OnTextFormatChanged Method

DiagramItem.OnTextFormatChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnTextFormatChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnTextFormatChanged ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.OnUpdateBounds Method

DiagramNode.OnUpdateBounds override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnUpdateBounds (
    bool sizeChanged
)
```

Visual Basic Copy Code

```
Protected Overrides Sub OnUpdateBounds( _
    sizeChanged As Boolean _
)
```

Parameters

sizeChanged
true if the size has changed; **false** if only the position has changed.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.OnZoomChanged Method

Called when the **DiagramView.ZoomFactor** changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnZoomChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnZoomChanged ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.RedimTable Method

Sets new number of rows and columns for this table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RedimTable (
    int cols,
    int rows
)
```

Visual Basic Copy Code

```
Public Sub RedimTable( _
    cols As Integer, _
    rows As Integer _
)
```

Parameters

cols
The new number of columns.

rows
The new number of rows.

Remarks

This method does not preserve the old contents of table cells. After its invocation all cells are empty and columns and rows are with default sizes.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
AddColumn Method
AddRow Method
DeleteColumn Method
DeleteRow Method

TableNode.ResizeToFitImage Method

Makes a table as big as the image it displays.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ResizeToFitImage ()
```

Visual Basic Copy Code

```
Public Sub ResizeToFitImage ()
```

Remarks

Call this method to make a table exactly the same size as the image it contains. The dots-per-inch resolution of the image is considered too when calculating the new table size. You can use ImageAlign to specify position of the picture and Image to set the picture itself.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
Image Property

TableNode.ResizeToFitText Method

Overload List

Name	Description
TableNode.ResizeToFitText (Boolean)	Resizes the columns and rows so that the cells are large enough to fit their text.
TableNode.ResizeToFitText (Boolean, Boolean)	Resizes the columns and rows so that the cells are large enough to fit their text.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.ResizeToFitText Method (Boolean)

Resizes the columns and rows so that the cells are large enough to fit their text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ResizeToFitText (
    bool ignoreCaption
)
```

Visual Basic Copy Code

```
Public Sub ResizeToFitText( _
    ignoreCaption As Boolean _
)
```

Parameters

ignoreCaption
false if the caption bar too should be resized to fit the caption text; otherwise, **true**.

Remarks

This method resizes the cells so they fit their Text on a single line, as long as the text does not contain new-line characters. The Width of each column is set to the width of its widest cell, and the Height of each row is set to the height of its highest cell. In the end, the table size is set to the sum of the extents of all rows and columns.

If *ignoreCaption* is **false**, the caption bar too will be resized, by setting CaptionHeight and the table width, so that it fits the Caption text.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
ResizeToFitImage Method

TableNode.ResizeToFitText Method (Boolean, Boolean)

Resizes the columns and rows so that the cells are large enough to fit their text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ResizeToFitText (
    bool ignoreCaption,
    bool keepCellWidth
)
```

Visual Basic Copy Code

```
Public Sub ResizeToFitText( _
    ignoreCaption As Boolean, _
    keepCellWidth As Boolean _
)
```

Parameters

ignoreCaption
false if the caption bar too should be resized to fit the caption text; otherwise, **true**.

keepCellWidth
true if column widths should stay unchanged; otherwise, **false**.

Remarks

This method resizes the cells so they fit their Text on a single line, as long as the text does not contain new-line characters. The Width of each column is set to the width of its widest cell, and the Height of each row is set to the height of its highest cell. In the end, the table size is set to the sum of the extents of all rows and columns.

If *ignoreCaption* is **false**, the caption bar too will be resized, by setting CaptionHeight and the table width, so that it fits the Caption text.

If *keepCellWidth* is **true**, the cell width is preserved unchanged.

See Also

ResizeToFitText Method Overload List
TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.RestoreProperties Method

DiagramNode.RestoreProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties -derived object containing a snapshot of the item's properties.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.SaveProperties Method

DiagramNode.SaveProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties -derived object where the item's properties should be saved.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.SaveToJson Method

Saves the node as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject instance to contain the serialized data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.SaveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** the item's data should be stored into.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.ScaleElements Method

DiagramNode.ScaleElements override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void ScaleElements (
    double scale
)
```

Visual Basic Copy Code

```
Protected Overrides Sub ScaleElements( _
    scale As Double _
)
```

Parameters

scale
The value by which to multiply node elements' sizes.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.ScrollDown Method

Scrolls the table rows downwards.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ScrollDown ()
```

Visual Basic Copy Code

```
Public Sub ScrollDown ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.ScrollUp Method

Scrolls the table rows upwards.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ScrollUp ()
```

Visual Basic Copy Code

```
Public Sub ScrollUp ()
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.SetReflexive Method

Invoked to let a node set the shape of a reflexive link connected to it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SetReflexive (
    DiagramLink link
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SetReflexive( _
    link As DiagramLink _
```

```
)
```

Parameters

link
Reference to the link.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode Properties

TableNode.AllowResizeColumns Property

Gets or sets a value indicating whether users are allowed to resize table columns.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowResizeColumns { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowResizeColumns As Boolean
```

Property Value

true to let users resize columns, or **false** otherwise. The default value is **false**.

Remarks

A column can be resized by dragging its right border. If the ColumnStyle property of a column is set to AutoWidth, it is automatically reset to **FixedWidth** when the column is resized. Rightmost columns always fill all remaining space in their tables and cannot be resized.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.AllowResizeRows Property

Gets or sets a value indicating whether users are allowed to resize table rows.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool AllowResizeRows { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowResizeRows As Boolean
```

Property Value

true to let users resize rows, or **false** otherwise. The default value is **false**.

Remarks

A row can be resized by dragging its bottom border.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.Caption Property

Gets or sets the text displayed as caption of the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Caption { get; set; }
```

Visual Basic Copy Code

```
Public Property Caption As String
```

Property Value

A string containing the text of the caption. Initialized with the value of TableCaption.

Remarks

Text color and alignment can be customized by using the CaptionBrush and CaptionFormat properties.

Styled text allows applying various attributes to the text characters. Attributes are specified by embedding HTML-like formatting tags within the raw text. Styled text is enabled by setting EnableStyledText to **true**.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
CaptionHeight Property
CaptionBackBrush Property
Diagram.TableCaption Property

TableNode.CaptionBackBrush Property

Gets or sets the brush used to paint the caption bar of this table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush CaptionBackBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionBackBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class. The default is **null** (**Nothing** in Visual Basic).

Remarks

This property refers to a brush object used to paint the caption bar. An instance of one of the following classes can be assigned to it: HatchBrush, LinearGradientBrush, PathGradientBrush, SolidBrush or TextureBrush, all located in the MindFusion.Drawing namespace. If there is no brush assigned to this property, the caption bar is painted using the Brush property of the TableNode class.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
CaptionHeight Property
Caption Property
DividerPen Property

TableNode.CaptionBrush Property

Gets or sets the brush used to paint the table caption.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush CaptionBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
CaptionHeight Property
CaptionFormat Property
Caption Property

TableNode.CaptionFormat Property

Gets or sets the alignment and line position of the text used as table caption.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringFormat CaptionFormat { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionFormat As StringFormat
```

Property Value

An instance of the StringFormat class.

Remarks

Use this property to change the alignment of text or its position inside the caption bar.

If there isn't enough space available to fit a whole line of text, MvcDiagram wraps the text to a new line when it reaches the edge of the table. You can specify that the text should be wrapped only at word boundaries by setting the **Trimming** member of **CaptionFormat** to **StringTrimming.Word**. The default is to split words into multiple parts if there is not enough space to fit a whole word.

If you prefer to use your own line-breaks, you can disable the automatic text wrapping by enabling the **NoWrap** flag in the **FormatFlags** member of **CaptionFormat**. If automatic wrapping is disabled and there isn't enough space to fit the whole line, the text is clipped at the table boundaries.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
CaptionHeight Property
CaptionBrush Property
Caption Property

TableNode.CaptionHeight Property

Gets or sets the height of the table's caption bar.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double CaptionHeight { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionHeight As Double
```

Property Value

A **double** value specifying the height of the table caption. Initialized with the value of TableCaptionHeight.

Remarks

Keep in mind that a table is moved by dragging its caption bar (per default HandlesStyle). Making the caption bar too narrow would make it harder for users to move.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
CaptionFormat Property
CaptionBrush Property
Caption Property
Diagram.TableCaptionHeight Property

TableNode.CaptionPlainText Property

Gets the plain text of the table's caption.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string CaptionPlainText { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property CaptionPlainText As String
```

Property Value

A string containing the plain text of the caption.

Remarks

This property removes any formatting tags from Caption and returns the plain human-readable text. That is useful when styled text is enabled and you need to display caption's content outside of MvcDiagram, that is, in a message box or a dialog box.

Example

The following sample code demonstrates how **CaptionPlainText** can be used.

C# Copy Code

```
t.EnableStyledText = true;
t.Caption = "<b>interface</b> <i><u>IMouseListener</u></i>";
MessageBox.Show(t.CaptionPlainText);
```

Visual Basic Copy Code

```
t.EnableStyledText = True
t.Caption = "<b>interface</b> <i><u>IMouseListener</u></i>"
MessageBox.Show(t.CaptionPlainText)
```

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
CaptionFormat Property
EnableStyledText Property

TableNode.CellCustomDraw Property

Gets or sets the type of custom drawing performed on the table cells.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CustomDraw CellCustomDraw { get; set; }
```

Visual Basic Copy Code

```
Public Property CellCustomDraw As CustomDraw
```

Property Value

A member of the CustomDraw enumeration. Initialized with the value of CellCustomDraw.

Remarks

This property allows invoking custom painting code for the cells of a table. That code must execute in response to the DrawCell event. There are several modes of custom painting, allowing the custom code to paint the cell entirely, or do that in addition to what MvcDiagram paints by default. The following CustomDraw members can be assigned to this property to specify the custom painting mode:

Member Name	Description
None	Do not perform custom drawing, let all painting be done by MvcDiagram.
Additional	The custom painting code executes after the cell interior is painted by MvcDiagram, but before the cell's text is rendered.
Full	The component does not do any standard drawing of the cell, the entire painting is left for the application.
Additional2	The custom painting code executes after all parts of the cell, including its text, are painted by MvcDiagram.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
TableNode.CustomDraw Property

TableNode.CellFrameStyle Property

Gets or sets the style of the table cells.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CellFrameStyle CellFrameStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property CellFrameStyle As CellFrameStyle
```

Property Value

A member of the CellFrameStyle enumeration. Initialized with the value of CellFrameStyle.

Remarks

This property specifies whether there are borders drawn around table cells, and if yes, their appearance.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
CellFrameStyle Enumeration
Diagram.CellFrameStyle Property

TableNode.CellTextBrush Property

Gets or sets the brush used to paint texts inside table cells.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush CellTextBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property CellTextBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.ColumnCount Property

Gets or sets the number of columns in the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int ColumnCount { get; set; }
```

Visual Basic Copy Code

```
Public Property ColumnCount As Integer
```

Property Value

An integer value indicating the number of columns in the table.

Remarks

If you increase the number of columns the contents of the old columns will remain intact. If you decrease the number, the last columns are removed and the other remain unchanged.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
Columns Property

TableNode.Columns Property

Gets a list of all columns in the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.ColumnCollection Columns { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Columns As TableNode.ColumnCollection
```

Property Value

An instance of the ColumnCollection class.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
Rows Property

TableNode.ColumnWidth Property

Gets or sets the default width of the table columns.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ColumnWidth { get; set; }
```

Visual Basic Copy Code

```
Public Property ColumnWidth As Double
```

Property Value

A **double** value specifying the width of the table columns. Initialized with the value of TableColumnWidth.

Remarks

The value of this property is used to initialize the width of new columns when they are created.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
RowHeight Property

TableNode.ConnectionStyle Property

Gets or sets a value indicating whether drawing a link from/to this table, should connect a row of the table, or the whole table as single entity.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableConnectionStyle ConnectionStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property ConnectionStyle As TableConnectionStyle
```

Property Value

A member of the TableConnectionStyle enumeration. Initialized with the value of TableConnectionStyle.

Remarks

There are three possible ways to relate tables to other nodes. This property specifies how users are allowed to create such relations:

- Rows - relation ends are the rows of a table. A link connected to the table is connected to a particular row. This mode is convenient for database diagrams.

- **Table** - relations target the table as a whole entity, and not any particular row. This mode is convenient for class diagrams.
- **Both** - a mixed mode where links can target either a row of the table or the table itself.

Programmatically, links can be connected to a table in any way, regardless of the **ConnectionStyle**. **ConnectionStyle** value is considered only when end-users draw links to or from a table. To link to a table as an entity and not to a specific row, pass -1 as row argument of the CreateDiagramLink method.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.CurrentScrollRow Property

Gets or sets the index of the first visible row in the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int CurrentScrollRow { get; set; }
```

Visual Basic Copy Code

```
Public Property CurrentScrollRow As Integer
```

Property Value

An integer value indicating the index of the first visible row.

Remarks

To allow users to scroll the table interactively, enable the Scrollable property.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
Rows Property

TableNode.CustomDraw Property

Gets or sets the type of custom drawing performed on this table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CustomDraw CustomDraw { get; set; }
```

Visual Basic Copy Code

```
Public Property CustomDraw As CustomDraw
```

Property Value

A member of the CustomDraw enumeration. Initialized with the value of TableCustomDraw.

Remarks

This property allows invoking custom painting code for a table. That code must execute in response to the DrawNode event. There are several modes of custom painting, allowing the custom code to paint the table entirely, or do that in addition to what

MvcDiagram paints by default. The following CustomDraw members can be assigned to this property to specify the custom painting mode:

Member Name	Description
None	Do not perform custom drawing, let all painting be done by MvcDiagram.
Additional	The custom painting code executes after the table interior and frame are painted by MvcDiagram, but before cells are rendered.
Full	The component does not do any standard drawing of the table frame, whose painting is left for the application. Cells are painted independently, as set in CellCustomDraw.
ShadowOnly	Only the shadow of the table is painted by custom code, the table itself is drawn by the component.
Additional2	The custom painting code executes after all parts of the table, including its cells, are painted by MvcDiagram.

This property doesn't affect the painting of table cells. Table cells can be custom painted in the DrawCell event as set in **CellCustomDraw**.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.EffectiveCaptionBackBrush Property

Gets the effective caption back brush of the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush EffectiveCaptionBackBrush { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectiveCaptionBackBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.EffectiveCaptionBrush Property

Gets the effective caption brush of the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush EffectiveCaptionBrush { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectiveCaptionBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.EffectiveCellTextBrush Property

Gets the effective text brush of the table cells.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush EffectiveCellTextBrush { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EffectiveCellTextBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.EnableStyledText Property

Gets or sets a value indicating whether styled text rendering is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableStyledText { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableStyledText As Boolean
```

Property Value

true to parse the text of table's caption and cells in order to find formatting tags; otherwise **false**. Initialized with the value of EnableStyledText.

Remarks

Styled text allows applying various attributes to the characters it contains. Attributes are specified using HTML-like formatting tags embedded in the raw text. Formatting tags are listed and described in the **Text Attributes and Formatting** section of this help file.

If **EnableStyledText** is set to **true**, table's Caption and cells' Text are parsed and analyzed each time they change. Parsing determines what formatting tags are used and adds the corresponding attribute records to MvcDiagram's internal representation of styled text. If the property value is **false**, the text is not processed in any way and is displayed without formatting.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
ResizeToFitText Method
CaptionPlainText Property

TableNode.Font Property

Gets or sets the font used when rendering the table's texts.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Font Font { get; set; }
```

Visual Basic Copy Code

```
Public Overrides Property Font As Font
```

Property Value

A Font instance specifying the font attributes.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.HasItems Property

DiagramNode.HasItems override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override bool HasItems { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property HasItems As Boolean
```

Property Value

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.HighlightedRow Property

Gets or sets the highlighted row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int HighlightedRow { get; set; }
```

Visual Basic Copy Code

```
Public Property HighlightedRow As Integer
```

Property Value

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.Image Property

Gets or sets the image that is displayed in this table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage Image { get; set; }
```

Visual Basic Copy Code

```
Public Property Image As IImage
```

Property Value

A **Microsoft.Maui.Graphics.IImage** instance.

Remarks

Use the property to customize your table's background. Further customization can be done with the ImageAlign property.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
ImageAlign Property
ImagePadding Property
ResizeToFitImage Method
Cell.Image Property

TableNode.ImageAlign Property

Gets or sets the alignment of the image displayed in this table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ImageAlign ImageAlign { get; set; }
```

Visual Basic Copy Code

```
Public Property ImageAlign As ImageAlign
```

Property Value

A member of the ImageAlign enumeration. The default value is **Fit**.

Remarks

This property specifies how images are positioned inside tables.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
Image Property
ImagePadding Property

TableNode.ImagePadding Property

Gets or sets the padding distance between node's borders and Image.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Nullable<Thickness> ImagePadding { get; set; }
```

Visual Basic Copy Code

```
Public Property ImagePadding As Nullable(Of Thickness)
```

Property Value

An optional Thickness value specifying padding distance; the default value is **null**.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
ImageAlign Property

TableNode.Item Property

Gets the Cell instance at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Cell Item [
    int col,
```

```
    int row
] { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Item( _
    col As Integer, _
    row As Integer _
) As TableNode.Cell
```

Parameters

col

The zero-based index of the cell's column.

row

The zero-based index of the cell's row.

Property Value

An instance of the Cell class that represents the cell of the table at the specified position.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
ColumnCount Property
RowCount Property
RedimTable Method

TableNode.OffsetHeaderRows Property

Gets or sets a value indicating whether the table cells are offset to the right to accommodate for the expand buttons of Header rows.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool OffsetHeaderRows { get; set; }
```

Visual Basic Copy Code

```
Public Property OffsetHeaderRows As Boolean
```

Property Value

true to offset table cells when there are header rows defined; otherwise, **false**. The default is **false**.

Remarks

If this property is set to **false**, the [±] buttons used to expand or collapse table sections are drawn inside the first cell of each header row. Otherwise, the buttons are drawn in their own column in front of the table's first column of cells.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.RowAnchorPattern Property

Gets or sets a default anchor pattern for the rows of this table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern RowAnchorPattern { get; set; }
```

Visual Basic Copy Code

```
Public Property RowAnchorPattern As AnchorPattern
```

Property Value

An instance of the AnchorPattern class. The default is **null** (**Nothing** in Visual Basic).

Remarks

Defines a default value for the AnchorPattern property of the rows of a table.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
AnchorPoint.Column Property
Row.AnchorPattern Property

TableNode.RowCount Property

Gets or sets the number of rows in the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int RowCount { get; set; }
```

Visual Basic Copy Code

```
Public Property RowCount As Integer
```

Property Value

An integer value specifying the number of rows in the table.

Remarks

If you increase the number of rows the content of the initial rows will remain intact. If you decrease the number, the last rows are removed and the other remain unchanged.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
Rows Property

TableNode.RowHeight Property

Gets or sets the default height of table rows.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double RowHeight { get; set; }
```

Visual Basic Copy Code

```
Public Property RowHeight As Double
```

Property Value

A **double** value specifying the height of the table rows. Initialized with the value of TableRowHeight.

Remarks

The value of this property is used to initialize the height of new rows when they are created.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
ColumnWidth Property

TableNode.Rows Property

Gets a list of all rows in the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.RowCollection Rows { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Rows As TableNode.RowCollection
```

Property Value

An instance of the RowCollection class.

Remarks

The property refers to the collection of all rows that compose the table. Do not add or remove rows through this collection, but use the table's AddRow, InsertRow or DeleteRow methods.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
Columns Property
CurrentScrollRow Property
Row.Table Property

TableNode.Scrollable Property

Gets or sets a value indicating whether users are allowed to scroll table rows.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Scrollable { get; set; }
```

Visual Basic Copy Code

```
Public Property Scrollable As Boolean
```

Property Value

true to allow scrolling the table up or down, or **false** otherwise. Initialized with the value of TablesScrollable.

Remarks

A button containing up and down arrows appears in the caption bar of a table for which scrolling is enabled. If a user clicks these arrows, table rows scroll up or down. The table can always be scrolled programmatically using the CurrentScrollRow property.

The appearance of the scroll icons can be customized by setting **Renderers.ScrollIconRenderer**.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace
Rows Property

TableNode.ScrollUI Property

Gets or sets what type of scroll UI to show in TableNodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static TableScrollUI ScrollUI { get; set; }
```

Visual Basic Copy Code

```
Public Shared Property ScrollUI As TableScrollUI
```

Property Value

A member of the TableScrollUI enumeration. The default value is **ScrollButtons**.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.Shape Property

Gets or sets the shape of the table's outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SimpleShape Shape { get; set; }
```

Visual Basic Copy Code

```
Public Property Shape As SimpleShape
```

Property Value

A member of the SimpleShape enumeration. Initialized with the value of TableShape.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.TextPadding Property

Gets or sets the spacing between the item boundaries and its text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Thickness TextPadding { get; set; }
```

Visual Basic Copy Code

```
Public Overrides Property TextPadding As Thickness
```

Property Value

A Thickness instance specifying the size of the padding space left around the text.

See Also

TableNode Members
TableNode Class
MindFusion.Diagramming Namespace

TableNode.Cell Class

Represents a cell in a TableNode instance.

Syntax

C# Copy Code

```
public class TableNode.Cell : InplaceEditable
```

Visual Basic Copy Code

```
Public Class TableNode.Cell
    Implements InplaceEditable
```

Remarks

Cells are the building blocks of tables. They get their height and width indirectly, from the width of table columns and the height of table rows as set by the ColumnWidth and RowHeight properties. Cells can be made to span several rows and columns by setting their RowSpan and ColumnSpan properties to values larger than 1.

Cells can hold text, which is set by their Text property. The text can be aligned and this is set by the TextFormat property. Color of the text in the cells is set with the TextBrush property.

Appearance of a cell can be customized further by the Brush and Image properties.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.TableNode.Cell

See Also

TableNode.Cell Members
MindFusion.Diagramming Namespace

TableNode.Cell Members

Represents a cell in a TableNode instance.

The following tables list the members exposed by the TableNode.Cell type.

Public Constructors

	Name	Description
	TableNode.Cell	Initializes a new instance of the **TableNode.Cell** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill this cell.
	ColumnSpan	Gets or sets how many columns the cell should span.
	Font	Gets or sets the font used to draw text inside this cell.
	HyperLink	Gets or sets the hyperlink associated with the cell.
	Image	Gets or sets the image that is displayed in this cell.
	ImageAlign	Gets or sets the alignment of the image displayed in this cell.
	ImagePadding	Gets or sets the padding distance between cell's borders and **Image**.
	ImageUrl	Gets or sets a URL used to load an image into this cell.
	PlainText	Gets the plain text of the cell.
	RowSpan	Gets or sets how many rows the cell should span.
	Table	Gets a reference to the **TableNode** containing this cell.
	Tag	Gets or sets custom data associated with this cell.
	Text	Gets or sets text displayed inside the cell.
	TextBrush	Gets or sets the brush used to paint the text of the cell.
	TextFormat	Gets or sets the alignment of the text that is displayed inside the cell.
	ToolTip	Gets or sets the tooltip text associated with the cell.

Public Methods

	Name	Description
	Clone	Creates a copy of this cell.

Protected Methods

	Name	Description
	LoadFromJson	Loads the cell from a JSON object.

	LoadFromXml	Loads the cell content from an XML element.
	SaveToJson	Saves the cell as a JSON object.
	SaveToXml	Saves the cell content into an XML element.

See Also

TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell Constructor

Initializes a new instance of the TableNode.Cell class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Cell (
    TableNode table
)
```

Visual Basic Copy Code

```
Public New ( _
    table As TableNode _
)
```

Parameters

table
The TableNode this cell belongs to.

See Also

TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell Methods

TableNode.Cell.Clone Method

Creates a copy of this cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Cell Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As TableNode.Cell
```

Return Value

An instance of the TableNode.Cell class.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell.LoadFromJson Method

Loads the cell from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the cell's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell.LoadFromXml Method

Loads the cell content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** containing the cell's data.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell.SaveToJson Method

Saves the cell as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject instance to contain the serialized data.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell.SaveToXml Method

Saves the cell content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** the cell's data should be stored into.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell Properties

TableNode.Cell.Brush Property

Gets or sets the brush used to fill this cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush Brush { get; set; }
```

Visual Basic Copy Code

```
Public Property Brush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class. The default is **null** (**Nothing** in Visual Basic).

Remarks

This property refers to a brush used to paint the cell's interior. You can assign to it an instance of one of the following classes: HatchBrush, LinearGradientBrush, PathGradientBrush, SolidBrush or TextureBrush, located in the MindFusion.Drawing namespace. If the default **null** value is used, the cell's interior is painted with the table's Brush instead of cell's one.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace
Image Property

TableNode.Cell.ColumnSpan Property

Gets or sets how many columns the cell should span.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int ColumnSpan { get; set; }
```

Visual Basic Copy Code

```
Public Property ColumnSpan As Integer
```

Property Value

A positive integer value specifying how many columns the cell spans to the right. The default is 1.

Remarks

Use this property and RowSpan to define a cell that overlaps several columns and rows of the table's grid. The spanning cell covers any non-spanning cells to the right or below it, as far as the spanned columns and rows go. The Image and Text of spanning cells are aligned and laid out inside the enlarged bounding rectangles of the cells. That allows composing nodes with a very elaborate structure using spanning cells, as alternative to attaching ShapeNode objects to a container/master node.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell.Font Property

Gets or sets the font used to draw text inside this cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Font Font { get; set; }
```

Visual Basic Copy Code

```
Public Property Font As Font
```

Property Value

A Font used to render the cell's Text, or **null** (**Nothing** in Visual Basic) if the text should be rendered using the table's Font.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell.HyperLink Property

Gets or sets the hyperlink associated with the cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string HyperLink { get; set; }
```

Visual Basic Copy Code

```
Public Property HyperLink As String
```

Property Value

A string value containing the hyperlink URL.

Remarks

This property is used by the CreateImageMap method of the HtmlBuilder class. Its value is assigned to the HREF attribute of the <AREA> tag that corresponds to the cell. Map areas are defined for table cells only if the AreasForTableCells property is enabled.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell.Image Property

Gets or sets the image that is displayed in this cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage Image { get; set; }
```

Visual Basic Copy Code

```
Public Property Image As IImage
```

Property Value

A **Microsoft.Maui.Graphics.IImage** instance. The default is **null** (**Nothing** in Visual Basic).

Remarks

Use this property to set an image to be displayed in the cell. Use the ImageAlign property to specify how the image is placed.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace
ImageAlign Property
ImagePadding Property

TableNode.Cell.ImageAlign Property

Gets or sets the alignment of the image displayed in this cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ImageAlign ImageAlign { get; set; }
```

Visual Basic Copy Code

```
Public Property ImageAlign As ImageAlign
```

Property Value

A member of the ImageAlign enumeration.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace
Image Property
ImagePadding Property

TableNode.Cell.ImagePadding Property

Gets or sets the padding distance between cell's borders and Image.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Nullable<Thickness> ImagePadding { get; set; }
```

Visual Basic Copy Code

```
Public Property ImagePadding As Nullable(Of Thickness)
```

Property Value

An optional Thickness value specifying padding distance; the default value is **null**.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace
ImageAlign Property

TableNode.Cell.ImageUrl Property

Gets or sets a URL used to load an image into this cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ImageUrl { get; set; }
```

Visual Basic Copy Code

```
Public Property ImageUrl As String
```

Property Value

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell.PlainText Property

Gets the plain text of the cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string PlainText { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property PlainText As String
```

Property Value

A string containing the plain text of the cell.

Remarks

This property removes any formatting tags from cell's Text and returns the plain human-readable text. That is useful when styled text is enabled and you need to display cell's content outside of MvcDiagram, that is, in a message or a dialog box.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace
TextFormat Property
TableNode.EnableStyledText Property

TableNode.Cell.RowSpan Property

Gets or sets how many rows the cell should span.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int RowSpan { get; set; }
```

Visual Basic Copy Code

```
Public Property RowSpan As Integer
```

Property Value

A positive integer value specifying how many rows the cell spans downwards. The default is 1.

Remarks

Use this property and ColumnSpan to define a cell that overlaps several columns and rows of the table's grid. The spanning cell covers any non-spanning cells to the right or below it, as far as the spanned columns and rows go. The Image and Text displayed in spanning cells are aligned and laid out inside the enlarged bounding rectangles of the cells. That allows composing nodes with a very elaborate structure using spanning cells, as an alternative to attaching ShapeNode objects to a container/master node.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell.Table Property

Gets a reference to the TableNode containing this cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode Table { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Table As TableNode
```

Property Value

An instance of the TableNode class representing the table containing this cell.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.Cell.Tag Property

Gets or sets custom data associated with this cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Tag { get; set; }
```

Visual Basic Copy Code

```
Public Property Tag As Object
```

Property Value

An instance of any class derived from **System.Object**. The default value is **null** (**Nothing** in Visual Basic).

Remarks

This property holds data that you define and is not used by MvcDiagram in any way. The tag could be any value of a primitive data type or a complex object from the .NET class library or a class defined by you. If the tag is an instance of a value type, it will be serialized together with the cell containing it when saving/loading the diagram. Reference types are serialized if marked with the **[Serializable]** attribute or if they implement the **ISerializable** interface.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace
Diagram.SerializeTag Event

TableNode.Cell.Text Property

Gets or sets text displayed inside the cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Text { get; set; }
```

Visual Basic Copy Code

```
Public Property Text As String
```

Property Value

A string value containing the text. The default is an empty string.

Remarks

The property specifies text that appears inside a cell. Text can be customized via the TextBrush and TextFormat properties.

Styled text allows applying various attributes to the text characters. Attributes are specified by embedding HTML-like formatting tags within the raw text. Styled text is enabled by setting EnableStyledText to **true**.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace
Image Property
TableNode.ResizeToFitText Method

TableNode.Cell.TextBrush Property

Gets or sets the brush used to paint the text of the cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush TextBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property TextBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class. The default is **null** (**Nothing** in Visual Basic).

Remarks

The Text of the cell is painted with this brush. However, if the property is set to the default **null** value, the table's CellTextBrush property is used instead of the cell's one.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace
Font Property
TextFormat Property

TableNode.Cell.TextFormat Property

Gets or sets the alignment of the text that is displayed inside the cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringFormat TextFormat { get; set; }
```

Visual Basic Copy Code

```
Public Property TextFormat As StringFormat
```

Property Value

An instance of the StringFormat class. The default alignment is **StringAlignment.Center** for lines and **StringAlignment.Near** for the written text.

Remarks

Use this property to specify the alignment of the text that is contained in cells.

If there isn't enough space available to fit a whole line of text, MvcDiagram wraps the text to a new line when it reaches the edge of the cell. You can specify that the text should be wrapped only at word boundaries by setting the **Trimming** member of **TextFormat** to **StringTrimming.Word**. The default is to split words into multiple parts if there is not enough space to fit a whole word.

If you prefer to use your own line-breaks, disable the automatic text wrapping by enabling the **NoWrap** flag in the **FormatFlags** member of **TextFormat**. If automatic wrapping is disabled and there isn't enough space to fit the whole line, the text is clipped at the cell boundaries.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace
Font Property
Text Property

TableNode.Cell.ToolTip Property

Gets or sets the tooltip text associated with the cell.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ToolTip { get; set; }
```

Visual Basic Copy Code

```
Public Property ToolTip As String
```

Property Value

The tool-tip text displayed for this cell. The default is an empty string.

Remarks

For more information, see the DiagramItem.ToolTip property.

See Also

TableNode.Cell Members
TableNode.Cell Class
MindFusion.Diagramming Namespace

TableNode.CellCollection Class

Represents a strongly-typed collection of TableNode.Cell objects.

Syntax

C# Copy Code

```
public class TableNode.CellCollection : CollectionBase
```

Visual Basic Copy Code

```
Public Class TableNode.CellCollection
    Inherits CollectionBase
```

Remarks

Do not use the methods and properties of this class to modify the contents of this collection directly. If you want to add or remove cells from a table, use the AddRow, AddColumn, InsertRow, InsertColumn, DeleteRow, DeleteColumn and RedimTable methods of the TableNode class instead.

Inheritance Hierarchy

System.Object
  System.Collections.CollectionBase
    MindFusion.Diagramming.TableNode.CellCollection

See Also

TableNode.CellCollection Members
MindFusion.Diagramming Namespace

TableNode.CellCollection Members

Represents a strongly-typed collection of TableNode.Cell objects.

The following tables list the members exposed by the TableNode.CellCollection type.

Public Constructors

	Name	Description
	TableNode.CellCollection	Initializes a new instance of the **TableNode.CellCollection** class.

Public Properties

	Name	Description
	Item	Gets or sets the cell at the specified position within the collection.

Public Methods

	Name	Description
	Add	Adds the specified cell to the collection.
	Clone	Creates a deep copy of this collection.
	GetAt	Returns the cell at the specified position.
	Insert	Inserts the specified cell at the specified index within the collection.
	Remove	Removes the specified cell from the collection.

See Also

TableNode.CellCollection Constructor

Initializes a new instance of the TableNode.CellCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.CellCollection ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

Remarks

This constructor supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

TableNode.CellCollection Class
MindFusion.Diagramming Namespace

TableNode.CellCollection Methods

TableNode.CellCollection.Add Method

Adds the specified cell to the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Add (
    TableNode.Cell cell
)
```

Visual Basic Copy Code

```
Public Sub Add( _
    cell As TableNode.Cell _
)
```

Parameters

cell
The TableNode.Cell object to add to the collection.

Remarks

Do not use this method directly. Use the methods of the TableNode class instead.

See Also

TableNode.CellCollection Members
TableNode.CellCollection Class

TableNode.CellCollection.Clone Method

Creates a deep copy of this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.CellCollection Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As TableNode.CellCollection
```

Return Value

An instance of the TableNode.CellCollection class.

Remarks

This method supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

TableNode.CellCollection Members
TableNode.CellCollection Class
MindFusion.Diagramming Namespace

TableNode.CellCollection.GetAt Method

Returns the cell at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Cell GetAt (
    int index
)
```

Visual Basic Copy Code

```
Public Function GetAt( _
    index As Integer _
) As TableNode.Cell
```

Parameters

index
The zero-based index of the cell to retrieve.

Return Value

A reference to the TableNode.Cell object at the specified position.

See Also

TableNode.CellCollection Members
TableNode.CellCollection Class
MindFusion.Diagramming Namespace

TableNode.CellCollection.Insert Method

Inserts the specified cell at the specified index within the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Insert (
    int index,
    TableNode.Cell cell
)
```

Visual Basic Copy Code

```
Public Sub Insert( _
    index As Integer, _
    cell As TableNode.Cell _
)
```

Parameters

index
The zero-based index at which to insert the cell.

cell
The TableNode.Cell object to add to the collection.

Remarks

Do not use this method directly. Use the methods of the TableNode class instead.

See Also

TableNode.CellCollection Members
TableNode.CellCollection Class
MindFusion.Diagramming Namespace

TableNode.CellCollection.Remove Method

Removes the specified cell from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Remove (
    TableNode.Cell cell
)
```

Visual Basic Copy Code

```
Public Sub Remove( _
    cell As TableNode.Cell _
)
```

Parameters

cell
The TableNode.Cell object to add to the collection.

Remarks

Do not use this method directly. Use the methods of the TableNode class instead.

See Also

TableNode.CellCollection Members
TableNode.CellCollection Class
MindFusion.Diagramming Namespace

TableNode.CellCollection Properties

TableNode.CellCollection.Item Property

Gets or sets the cell at the specified position within the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Cell Item [
    int index
] { get; set; }
```

Visual Basic Copy Code

```
Public Property Item( _
    index As Integer _
) As TableNode.Cell
```

Parameters

index
The zero-based index of the item which should be get or set.

Property Value

A reference to the TableNode.Cell object at the specified index.

Remarks

Do not use the *set* accessor of this property. It supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

TableNode.CellCollection Members
TableNode.CellCollection Class
MindFusion.Diagramming Namespace

TableNode.Column Class

Represents a column of a TableNode.

Syntax

C# Copy Code

```
public class TableNode.Column
```

Visual Basic Copy Code

```
Public Class TableNode.Column
```

Remarks

The width of a column can be set via its Width property. The width of a column determines the width of all cells contained within the column.

When resizing a table, its columns can be resized too. That depends on the value of ColumnStyle, which specifies whether columns are with fixed width or they resize to keep column-to-table width ratio.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.TableNode.Column

See Also

TableNode.Column Members
MindFusion.Diagramming Namespace

TableNode.Column Members

Represents a column of a TableNode.

The following tables list the members exposed by the TableNode.Column type.

Public Constructors

	Name	Description
	TableNode.Column	Initializes a new instance of the **TableNode.Column** class.

Public Properties

	Name	Description
	ColumnStyle	Gets or sets the manner in which column's width is set.
	Width	Gets or sets the width of the table column.

Public Methods

	Name	Description
	Clone	Creates a copy of this column.

Protected Methods

	Name	Description
	LoadFromJson	Loads the column from a JSON object.
	LoadFromXml	Loads the column properties from an XML element.
	SaveToJson	Saves the column as a JSON object.
	SaveToXml	Saves the column properties into an XML element.

See Also

TableNode.Column Class
MindFusion.Diagramming Namespace

TableNode.Column Constructor

Initializes a new instance of the TableNode.Column class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Column (
    TableNode tableNode
)
```

Visual Basic Copy Code

```
Public New ( _
    tableNode As TableNode _
)
```

Parameters

tableNode
The TableNode this column belongs to.

See Also

TableNode.Column Class
MindFusion.Diagramming Namespace

TableNode.Column Methods

TableNode.Column.Clone Method

Creates a copy of this column.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Column Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As TableNode.Column
```

Return Value

An instance of the TableNode.Column class.

See Also

TableNode.Column Members
TableNode.Column Class
MindFusion.Diagramming Namespace

TableNode.Column.LoadFromJson Method

Loads the column from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the column's data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

TableNode.Column Members
TableNode.Column Class
MindFusion.Diagramming Namespace

TableNode.Column.LoadFromXml Method

Loads the column properties from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** containing the column's data.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

TableNode.Column Members
TableNode.Column Class
MindFusion.Diagramming Namespace

TableNode.Column.SaveToJson Method

Saves the column as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject instance to contain the serialized data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

TableNode.Column Members
TableNode.Column Class
MindFusion.Diagramming Namespace

TableNode.Column.SaveToXml Method

Saves the column properties into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** the column's data should be stored into.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

TableNode.Column Members
TableNode.Column Class
MindFusion.Diagramming Namespace

TableNode.Column Properties

TableNode.Column.ColumnStyle Property

Gets or sets the manner in which column's width is set.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ColumnStyle ColumnStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property ColumnStyle As ColumnStyle
```

Property Value

A member of the ColumnStyle enumeration. The default is **ColumnStyle.FixedWidth**.

Remarks

This property is used for calculating the width of a table column. The column can have either a fixed width set through the Width property, or can be resized proportionally to the table width.

See Also

TableNode.Column Members
TableNode.Column Class
MindFusion.Diagramming Namespace

TableNode.Column.Width Property

Gets or sets the width of the table column.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Width { get; set; }
```

Visual Basic Copy Code

```
Public Property Width As Double
```

Property Value

A **double** value indicating the width of the column. The default is 50.

See Also

TableNode.Column Members
TableNode.Column Class
MindFusion.Diagramming Namespace
TableNode.ColumnWidth Property

TableNode.ColumnCollection Class

Represents a strongly-typed collection of TableNode.Column objects.

Syntax

C# Copy Code

```
public class TableNode.ColumnCollection : CollectionBase
```

Visual Basic Copy Code

```
Public Class TableNode.ColumnCollection
    Inherits CollectionBase
```

Remarks

Do not use the methods and properties of this class to modify the contents of this collection directly. If you want to add or remove columns from a table, use the AddColumn, InsertColumn, DeleteColumn and RedimTable methods of the TableNode class instead.

Inheritance Hierarchy

System.Object
  System.Collections.CollectionBase
    MindFusion.Diagramming.TableNode.ColumnCollection

See Also

TableNode.ColumnCollection Members
MindFusion.Diagramming Namespace

TableNode.ColumnCollection Members

Represents a strongly-typed collection of TableNode.Column objects.

The following tables list the members exposed by the TableNode.ColumnCollection type.

Public Constructors

	Name	Description
	TableNode.ColumnCollection	Initializes a new instance of the **TableNode.ColumnCollection** class.

Public Properties

	Name	Description
	Item	Gets or sets the column at the specified position within the collection.

Public Methods

	Name	Description
	Add	Adds the specified column to the collection.
	Clone	Creates a deep copy of this collection.
	GetAt	Returns the column at the specified position.
	Insert	Inserts the specified column at the specified index within the collection.
	Remove	Removes the specified column from the collection.

See Also

TableNode.ColumnCollection Class
MindFusion.Diagramming Namespace

TableNode.ColumnCollection Constructor

Initializes a new instance of the TableNode.ColumnCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.ColumnCollection ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

Remarks

This constructor supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

TableNode.ColumnCollection Class
MindFusion.Diagramming Namespace

TableNode.ColumnCollection Methods

TableNode.ColumnCollection.Add Method

Adds the specified column to the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Add (
    TableNode.Column column
)
```

Visual Basic Copy Code

```
Public Sub Add( _
    column As TableNode.Column _
)
```

Parameters

column
The TableNode.Column object to add to the collection.

Remarks

Do not use this method directly. Use the AddColumn method of the TableNode class instead.

See Also

TableNode.ColumnCollection Members
TableNode.ColumnCollection Class
MindFusion.Diagramming Namespace

TableNode.ColumnCollection.Clone Method

Creates a deep copy of this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.ColumnCollection Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As TableNode.ColumnCollection
```

Return Value

An instance of the TableNode.ColumnCollection class.

Remarks

This method supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

TableNode.ColumnCollection Members
TableNode.ColumnCollection Class
MindFusion.Diagramming Namespace

TableNode.ColumnCollection.GetAt Method

Returns the column at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Column GetAt (
    int index
)
```

Visual Basic Copy Code

```
Public Function GetAt( _
    index As Integer _
) As TableNode.Column
```

Parameters

index

The zero-based index of the column to retrieve.

Return Value

A reference to the TableNode.Column object at the specified position.

See Also

TableNode.ColumnCollection Members
TableNode.ColumnCollection Class
MindFusion.Diagramming Namespace

TableNode.ColumnCollection.Insert Method

Inserts the specified column at the specified index within the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Insert (
    int index,
    TableNode.Column column
)
```

Visual Basic Copy Code

```
Public Sub Insert( _
    index As Integer, _
    column As TableNode.Column _
)
```

Parameters

index
The zero-based index at which to insert the column.

column
The TableNode.Column object to add to the collection.

Remarks

Do not use this method directly. Use the InsertColumn method of the TableNode class instead.

See Also

TableNode.ColumnCollection Members
TableNode.ColumnCollection Class
MindFusion.Diagramming Namespace

TableNode.ColumnCollection.Remove Method

Removes the specified column from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Remove (
    TableNode.Column column
)
```

Visual Basic Copy Code

```
Public Sub Remove( _
    column As TableNode.Column _
)
```

Parameters

column
The TableNode.Column object to add to the collection.

Remarks

Do not use this method directly. Use the DeleteColumn method of the TableNode class instead.

See Also

TableNode.ColumnCollection Members
TableNode.ColumnCollection Class
MindFusion.Diagramming Namespace

TableNode.ColumnCollection Properties

TableNode.ColumnCollection.Item Property

Gets or sets the column at the specified position within the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Column Item [
    int index
] { get; set; }
```

Visual Basic Copy Code

```
Public Property Item( _
    index As Integer _
) As TableNode.Column
```

Parameters

index
The zero-based index of the item which should be get or set.

Property Value

A reference to the TableNode.Column object at the specified index.

Remarks

Do not use the *set* accessor of this property. It supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

TableNode.ColumnCollection Members
TableNode.ColumnCollection Class
MindFusion.Diagramming Namespace

TableNode.Row Class

Represents a row in TableNode node.

Syntax

C# Copy Code

```
public class TableNode.Row
```

Visual Basic Copy Code

```
Public Class TableNode.Row
```

Remarks

Table rows can be assigned to distinct sections of a table. For example, sections could contain rows representing different types of class members in class diagrams, there could be one section displaying the methods, another one for the events and one more for the properties of a class. Each section can be collapsed or expanded, hiding or showing all rows in the section except the header one. To define a section, set a Header row. A section consists of all rows after a header row and spans to the next header. Each header row displays a [±] button that allows expanding or collapsing the section interactively. Sections can be collapsed and expanded programmatically using the Expanded property of the header row.

The height of a row is set via the Height property. That determines the height of all non-spanning cells contained within the row. Cells can be made to span several rows by setting their RowSpan property.

Rows can be bound together to represent relationship between database tables. The relationship can be one to many and many to one. The tables that stay on the left side of M:1 relationships are stored in the RelatedManyToOne collection; the tables in 1:M relationship with the row are stored in the RelatedOneToMany collection.

The relationship between tables is shown with links. You can access the DiagramLink elements that go out of a table row via the OutgoingLinks property. Links that come into a given row can be accessed with the IncomingLinks property.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.TableNode.Row

See Also

TableNode.Row Members
MindFusion.Diagramming Namespace

TableNode.Row Members

Represents a row in TableNode node.

The following tables list the members exposed by the TableNode.Row type.

Public Constructors

	Name	Description
	TableNode.Row	Initializes a new instance of the **TableNode.Row** class.

Public Properties

	Name	Description
	AnchorPattern	Gets or sets a set of anchor points to which links can dock when connected to this row.
	Expanded	Gets or sets a value indicating whether the table section starting from this Header row is collapsed or expanded.
	Header	Gets or sets a value indicating whether this row is a section header.
	Height	Gets or sets the height of the table row.
	IncomingLinks	Gets a collection of all links that point to this table row.
	OutgoingLinks	Gets a collection of all links that go out of this table row.
	RelatedManyToOne	Gets a collection of all tables that are in many-to-one relation with this table row.
	RelatedOneToMany	Gets a collection of all tables that are in one-to-many relation with this table row.
	Table	Gets the **TableNode** that contains this row.

Public Methods

	Name	Description
	Clone	Creates a copy of this row.

Protected Methods

	Name	Description
	LoadFromJson	Loads the row from a JSON object.
	LoadFromXml	Loads the row properties from an XML element.
	SaveToJson	Saves the row as a JSON object.
	SaveToXml	Saves the row properties into an XML element.

See Also

TableNode.Row Class
MindFusion.Diagramming Namespace

TableNode.Row Constructor

Initializes a new instance of the TableNode.Row class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Row (
    TableNode table
)
```

Visual Basic Copy Code

```
Public New ( _
    table As TableNode _
)
```

Parameters

table
The TableNode this row belongs to.

See Also

TableNode.Row Class
MindFusion.Diagramming Namespace

TableNode.Row Methods

TableNode.Row.Clone Method

Creates a copy of this row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Row Clone ()
```

Visual Basic

```
Public Function Clone () As TableNode.Row
```

Return Value

An instance of the TableNode.Row class.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace

TableNode.Row.LoadFromJson Method

Loads the row from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic

```
Protected Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject containing the row's data.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace

TableNode.Row.LoadFromXml Method

Loads the row properties from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
protected void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic

```
Protected Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** containing the row's data.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace

TableNode.Row.SaveToJson Method

Saves the row as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject instance to contain the serialized data.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace

TableNode.Row.SaveToXml Method

Saves the row properties into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** the row's data should be stored into.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace

TableNode.Row Properties

TableNode.Row.AnchorPattern Property

Gets or sets a set of anchor points to which links can dock when connected to this row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern AnchorPattern { get; set; }
```

Visual Basic Copy Code

```
Public Property AnchorPattern As AnchorPattern
```

Property Value

An instance of the AnchorPattern class.

Remarks

AnchorPattern defines a set of locations where links connected to a row are allowed to connect. When a link is connected to a row, the respective end of the link is aligned to the nearest AnchorPoint assigned to that row. If a row does not have associated anchor points of the appropriate type (incoming/outgoing), the link will be connected to the nearest row that has such anchors defined. If none of the rows of a table has an associated anchor pattern, links are connected to the middle of the left or right side of the pointed row.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace
TableNode.RowAnchorPattern Property
AnchorPoint.Column Property

TableNode.Row.Expanded Property

Gets or sets a value indicating whether the table section starting from this Header row is collapsed or expanded.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Expanded { get; set; }
```

Visual Basic Copy Code

```
Public Property Expanded As Boolean
```

Property Value

true if the table section that starts from this row is expanded; otherwise, **false**. The default is **true**.

Remarks

This property can be set only if the row is a Header. Links connected to the rows of a collapsed table section appear to be connected to the section header row.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace

TableNode.Row.Header Property

Gets or sets a value indicating whether this row is a section header.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Header { get; set; }
```

Visual Basic Copy Code

```
Public Property Header As Boolean
```

Property Value

true if this is the header row of a table section, otherwise **false**. The default is **false**.

Remarks

Each header row defines a section of the table that includes all non-header rows after the header. Such sections can be expanded or collapsed, that is, the rows included in them can be shown or hidden. This can be done interactively by pressing the [±] button displayed in header rows, or programmatically by setting the Expanded property. The TableSectionCollapsed and TableSectionExpanded events are raised when a user clicks the [±] button.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace
OffsetHeaderRows Property

TableNode.Row.Height Property

Gets or sets the height of the table row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Height { get; set; }
```

Visual Basic Copy Code

```
Public Property Height As Double
```

Property Value

A **double** value indicating the height of the row. The default is 22.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace
TableNode.RowHeight Property

TableNode.Row.IncomingLinks Property

Gets a collection of all links that point to this table row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection IncomingLinks { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property IncomingLinks As DiagramLinkCollection
```

Property Value

A DiagramLinkCollection containing references to the incoming links.

Remarks

The property returns a collection of all links that point to this table row. Use it to access the links or get their number.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace
OutgoingLinks Property

TableNode.Row.OutgoingLinks Property

Gets a collection of all links that go out of this table row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection OutgoingLinks { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property OutgoingLinks As DiagramLinkCollection
```

Property Value

A DiagramLinkCollection containing references to the outgoing links.

Remarks

The property returns a collection of all links that start from this table row. Use it to access the links or get their number.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace
IncomingLinks Property

TableNode.Row.RelatedManyToOne Property

Gets a collection of all tables that are in many-to-one relation with this table row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeCollection RelatedManyToOne { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property RelatedManyToOne As DiagramNodeCollection
```

Property Value

A DiagramNodeCollection containing references to the TableNode objects, which are in relation M:1 with this row.

Remarks

When working with database diagrams, use this property to find all tables which are in M:1 relation with this row.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace
RelatedOneToMany Property

TableNode.Row.RelatedOneToMany Property

Gets a collection of all tables that are in one-to-many relation with this table row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramNodeCollection RelatedOneToMany { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property RelatedOneToMany As DiagramNodeCollection
```

Property Value

A DiagramNodeCollection containing references to the TableNode objects, which are in relation 1:M with this row.

Remarks

When working with database diagrams, use this property to find all tables which are in 1:M relation with this row.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace

TableNode.Row.Table Property

Gets the TableNode that contains this row.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode Table { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Table As TableNode
```

Property Value

A TableNode instance.

See Also

TableNode.Row Members
TableNode.Row Class
MindFusion.Diagramming Namespace

TableNode.RowCollection Class

Represents a strongly-typed collection of TableNode.Row objects.

Syntax

C# Copy Code

```
public class TableNode.RowCollection : CollectionBase
```

Visual Basic Copy Code

```
Public Class TableNode.RowCollection
    Inherits CollectionBase
```

Remarks

Do not use the methods and properties of this class to modify the contents of this collection directly. If you want to add or remove rows from a table, use the AddRow, InsertRow, DeleteRow and RedimTable methods of the TableNode class instead.

Inheritance Hierarchy

System.Object
  System.Collections.CollectionBase
    MindFusion.Diagramming.TableNode.RowCollection

See Also

TableNode.RowCollection Members
MindFusion.Diagramming Namespace

TableNode.RowCollection Members

Represents a strongly-typed collection of TableNode.Row objects.

The following tables list the members exposed by the TableNode.RowCollection type.

Public Constructors

	Name	Description
	TableNode.RowCollection	Initializes a new instance of the **TableNode.RowCollection** class.

Public Properties

	Name	Description
	Item	Gets or sets the row at the specified position within the collection.

Public Methods

	Name	Description
	Add	Adds the specified row to the collection.
	Clone	Creates a deep copy of this collection.
	GetAt	Returns the row at the specified position.
	IndexOf	Returns the index of the specified row within the collection.
	Insert	Inserts the specified row at the specified index within the collection.
	Remove	Removes the specified row from the collection.

See Also

TableNode.RowCollection Class
MindFusion.Diagramming Namespace

TableNode.RowCollection Constructor

Initializes a new instance of the TableNode.RowCollection class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.RowCollection ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

Remarks

This constructor supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

TableNode.RowCollection Class
MindFusion.Diagramming Namespace

TableNode.RowCollection Methods

TableNode.RowCollection.Add Method

Adds the specified row to the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Add (
    TableNode.Row r
)
```

Visual Basic Copy Code

```
Public Sub Add( _
    r As TableNode.Row _
)
```

Parameters

r
The TableNode.Row object to add to the collection.

Remarks

Do not use this method directly. Use the AddRow method of the TableNode class instead.

See Also

TableNode.RowCollection Members
TableNode.RowCollection Class
MindFusion.Diagramming Namespace

TableNode.RowCollection.Clone Method

Creates a deep copy of this collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.RowCollection Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As TableNode.RowCollection
```

Return Value

An instance of the TableNode.RowCollection class.

Remarks

This method supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

TableNode.RowCollection Members
TableNode.RowCollection Class
MindFusion.Diagramming Namespace

TableNode.RowCollection.GetAt Method

Returns the row at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Row GetAt (
    int index
)
```

Visual Basic Copy Code

```
Public Function GetAt( _
    index As Integer _
) As TableNode.Row
```

Parameters

index
The zero-based index of the row to retrieve.

Return Value

A reference to the TableNode.Row object at the specified position.

Remarks

A reference to the TableNode.Row object at the specified position.

See Also

TableNode.RowCollection Members
TableNode.RowCollection Class
MindFusion.Diagramming Namespace

TableNode.RowCollection.IndexOf Method

Returns the index of the specified row within the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int IndexOf (
    TableNode.Row row
```

```
)
```

Visual Basic Copy Code

```
Public Function IndexOf( _
    row As TableNode.Row _
) As Integer
```

Parameters

row
The TableNode.Row object whose index to search for in the collection.

Return Value

An integer value specifying the index of the row within the collection or -1, if the row was not found.

See Also

TableNode.RowCollection Members
TableNode.RowCollection Class
MindFusion.Diagramming Namespace

TableNode.RowCollection.Insert Method

Inserts the specified row at the specified index within the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Insert (
    int index,
    TableNode.Row row
)
```

Visual Basic Copy Code

```
Public Sub Insert( _
    index As Integer, _
    row As TableNode.Row _
)
```

Parameters

index
The zero-based index at which to insert the row.

row
The TableNode.Row object to add to the collection.

Remarks

Do not use this method directly. Use the InsertRow method of the TableNode class instead.

See Also

TableNode.RowCollection Members
TableNode.RowCollection Class
MindFusion.Diagramming Namespace

TableNode.RowCollection.Remove Method

Removes the specified row from the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Remove (
    TableNode.Row row
)
```

Visual Basic Copy Code

```
Public Sub Remove( _
    row As TableNode.Row _
)
```

Parameters

row
The TableNode.Row object to add to the collection.

Remarks

Do not use this method directly. Use the DeleteRow method of the TableNode class instead.

See Also

TableNode.RowCollection Members
TableNode.RowCollection Class
MindFusion.Diagramming Namespace

TableNode.RowCollection Properties

TableNode.RowCollection.Item Property

Gets or sets the row at the specified position within the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNode.Row Item [
    int index
] { get; set; }
```

Visual Basic Copy Code

```
Public Property Item( _
    index As Integer _
) As TableNode.Row
```

Parameters

index

Property Value

A reference to the TableNode.Row object at the specified index.

Remarks

Do not use the *set* accessor of this property. It supports the MvcDiagram infrastructure and is not intended to be used directly from your code.

See Also

TableNodeProperties Class

Stores the values of TableNode properties during undo / redo operations.

Syntax

C# Copy Code

```
public class TableNodeProperties : DiagramNodeProperties
```

Visual Basic Copy Code

```
Public Class TableNodeProperties
    Inherits DiagramNodeProperties
```

Remarks

If you need to implement undo support for the additional properties defined by a class derived from TableNode, do so by deriving from **TableNodeProperties** and overriding the SaveProperties, RestoreProperties, and CreateProperties methods of **TableNode**.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItemProperties
    MindFusion.Diagramming.DiagramNodeProperties
      MindFusion.Diagramming.TableNodeProperties

See Also

TableNodeProperties Members
MindFusion.Diagramming Namespace

TableNodeProperties Members

Stores the values of TableNode properties during undo / redo operations.

The following tables list the members exposed by the TableNodeProperties type.

Public Constructors

	Name	Description
	TableNodeProperties	Initializes a new instance of the **TableNodeProperties** class.

Public Fields

	Name	Description
	Caption	For internal use.
	CaptionHeight	For internal use.
	CellTextBrush	For internal use.
	ColumnWidth	For internal use.
	CurrScrollRow	For internal use.
	ItemClass	For internal use. (Inherited from DiagramItemProperties.)
	labels	For internal use. (Inherited from DiagramNodeProperties.)

	RowHeight	For internal use.
	Scrollable	For internal use.
	Selected	For internal use. (Inherited from DiagramItemProperties.)
	Text	For internal use. (Inherited from DiagramItemProperties.)
	TextPadding	For internal use. (Inherited from DiagramItemProperties.)
	ZIndex	For internal use. (Inherited from DiagramItemProperties.)

Public Methods

	Name	Description
	FreeResources	DiagramItemProperties.FreeResources override.

See Also

TableNodeProperties Class
MindFusion.Diagramming Namespace

TableNodeProperties Fields

TableNodeProperties.Caption Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Caption
```

Visual Basic Copy Code

```
Public Caption As String
```

See Also

TableNodeProperties Members
TableNodeProperties Class
MindFusion.Diagramming Namespace

TableNodeProperties.CaptionHeight Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double CaptionHeight
```

Visual Basic Copy Code

```
Public CaptionHeight As Double
```

See Also

TableNodeProperties Members
TableNodeProperties Class
MindFusion.Diagramming Namespace

TableNodeProperties.CellTextBrush Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush CellTextBrush
```

Visual Basic Copy Code

```
Public CellTextBrush As Brush
```

See Also

TableNodeProperties Members
TableNodeProperties Class
MindFusion.Diagramming Namespace

TableNodeProperties.ColumnWidth Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ColumnWidth
```

Visual Basic Copy Code

```
Public ColumnWidth As Double
```

See Also

TableNodeProperties Members
TableNodeProperties Class
MindFusion.Diagramming Namespace

TableNodeProperties.CurrScrollRow Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int CurrScrollRow
```

Visual Basic Copy Code

```
Public CurrScrollRow As Integer
```

See Also

TableNodeProperties Members
TableNodeProperties Class
MindFusion.Diagramming Namespace

TableNodeProperties.RowHeight Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double RowHeight
```

Visual Basic Copy Code

```
Public RowHeight As Double
```

See Also

TableNodeProperties Members
TableNodeProperties Class
MindFusion.Diagramming Namespace

TableNodeProperties.Scrollable Field

For internal use.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Scrollable
```

Visual Basic Copy Code

```
Public Scrollable As Boolean
```

See Also

TableNodeProperties Members
TableNodeProperties Class
MindFusion.Diagramming Namespace

TableNodeProperties Constructor

Initializes a new instance of the TableNodeProperties class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNodeProperties ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

TableNodeProperties Class
MindFusion.Diagramming Namespace

TableNodeProperties Methods

TableNodeProperties.FreeResources Method

DiagramItemProperties.FreeResources override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void FreeResources ()
```

Visual Basic Copy Code

```
Public Overrides Sub FreeResources ()
```

See Also

TableNodeProperties Members
TableNodeProperties Class
MindFusion.Diagramming Namespace

TableNodeStyle Class

Represents a reusable set of properties that define the appearance of table nodes.

Syntax

C# Copy Code

```
public class TableNodeStyle : DiagramNodeStyle
```

Visual Basic Copy Code

```
Public Class TableNodeStyle
    Inherits DiagramNodeStyle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle
      MindFusion.Diagramming.DiagramNodeStyle
        MindFusion.Diagramming.TableNodeStyle

See Also

TableNodeStyle Members
MindFusion.Diagramming Namespace

TableNodeStyle Members

Represents a reusable set of properties that define the appearance of table nodes.

The following tables list the members exposed by the TableNodeStyle type.

Public Constructors

	Name	Description
	TableNodeStyle	Initializes a new instance of the **TableNodeStyle** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of the item. (Inherited from DiagramItemStyle.)
	CaptionBackBrush	Gets or sets the brush used to paint the caption bar of a table.
	CellTextBrush	Gets or sets the brush used to paint texts inside table cells.
	Effects	Gets a list of the effects applied to the node. (Inherited from DiagramNodeStyle.)
	FontFamily	Gets or sets the font family of the item text. (Inherited from DiagramItemStyle.)
	FontSize	Gets or sets the font size of the item text. (Inherited from DiagramItemStyle.)
	FontStyle	Gets or sets the font style of the item text. (Inherited from DiagramItemStyle.)
	FontUnit	Gets or sets the font unit of the item text. (Inherited from DiagramItemStyle.)
	ShadowBrush	Gets or sets the brush used to paint the shadow of the item. (Inherited from DiagramItemStyle.)
	Stroke	Gets or sets the brush used to paint the outline of the item. (Inherited from DiagramItemStyle.)
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items. (Inherited from DiagramItemStyle.)
	StrokeThickness	Gets or sets the thickness of the item's outline, expressed in the current units of measure. (Inherited from DiagramItemStyle.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItemStyle.)

Public Methods

	Name	Description

	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
S	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JsonObject. (Inherited from DiagramNodeStyle.)
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)
	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a JsonObject. (Inherited from DiagramNodeStyle.)
	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

TableNodeStyle Class
MindFusion.Diagramming Namespace

TableNodeStyle Constructor

Initializes a new instance of the TableNodeStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TableNodeStyle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

TableNodeStyle Properties

TableNodeStyle.CaptionBackBrush Property

Gets or sets the brush used to paint the caption bar of a table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush CaptionBackBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionBackBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

TableNodeStyle Members
TableNodeStyle Class
MindFusion.Diagramming Namespace

TableNodeStyle.CellTextBrush Property

Gets or sets the brush used to paint texts inside table cells.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush CellTextBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property CellTextBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class.

See Also

TableNodeStyle Members
TableNodeStyle Class
MindFusion.Diagramming Namespace

TableNodeStyle Events

TableScrollUI Enumeration

Identifies types of scroll UI supported by TableNode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum TableScrollUI
```

Visual Basic Copy Code

```
Public Enum TableScrollUI
```

Members

	Member name	Description
	Scrollbar	Let users scroll using a scrollbar show on the right side of a TableNode.
	ScrollButtons	Show scroll-up and scroll-down arrow buttons in table's caption bar.

See Also

MindFusion.Diagramming Namespace

Theme Class

Represents a set of Style objects for different diagram item types.

Syntax

C# Copy Code

```
public class Theme
```

Visual Basic Copy Code

```
Public Class Theme
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Theme

See Also

Theme Members
MindFusion.Diagramming Namespace

Theme Members

Represents a set of Style objects for different diagram item types.

The following tables list the members exposed by the Theme type.

Public Constructors

	Name	Description

	Theme	Initializes a new instance of the **Theme** class.

Public Methods

	Name	Description
	GetStyle	Gets the Style object, if any, associated with the specified type.
	LoadFromString	Loads the theme from an XML string.
	LoadFromXml	Loads the theme from an XML file.
	RegisterStyle	Registers the specified **Style** object for the items of the specified type.
	SaveToString	Saves the theme into an XML string.
	SaveToXml	Saves the theme to an XML file.
	UnregisterStyle	Unregisters the **Style** object associated with the specified type.

See Also

Theme Class
MindFusion.Diagramming Namespace

Theme Constructor

Initializes a new instance of the Theme class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Theme ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

Theme Class
MindFusion.Diagramming Namespace

Theme Methods

Theme.GetStyle Method

Gets the Style object, if any, associated with the specified type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Style GetStyle (
    Type type
)
```

Visual Basic

```
Public Function GetStyle( _
    type As Type _
) As Style
```

Parameters

type
The runtime type whose style to retrieve.

Return Value

The Style associated with the type, or null.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.LoadFromString Method

Loads the theme from an XML string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void LoadFromString (
    string xml
)
```

Visual Basic

```
Public Sub LoadFromString( _
    xml As String _
)
```

Parameters

xml
The string to load from.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.LoadFromXml Method

Loads the theme from an XML file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void LoadFromXml (
    string fileName
)
```

Visual Basic

```
Public Sub LoadFromXml( _
    fileName As String _
)
```

Parameters

fileName
The name of the file to load from.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.RegisterStyle Method

Registers the specified Style object for the items of the specified type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void RegisterStyle (
    Type type,
    Style style
)
```

Visual Basic

```
Public Sub RegisterStyle( _
    type As Type, _
    style As Style _
)
```

Parameters

type
The runtime type to register.

style
The style to associate with the specified type. The previously associated style, if any, is discarded.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.SaveToString Method

Saves the theme into an XML string.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public string SaveToString ()
```

Visual Basic

```
Public Function SaveToString () As String
```

Return Value

The serialized string.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.SaveToXml Method

Saves the theme to an XML file.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void SaveToXml (
    string fileName
)
```

Visual Basic

```
Public Sub SaveToXml( _
    fileName As String _
)
```

Parameters

fileName
The name of the file to save to.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Theme.UnregisterStyle Method

Unregisters the Style object associated with the specified type.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public void UnregisterStyle (
    Type type
)
```

Visual Basic

```
Public Sub UnregisterStyle( _
    type As Type _
)
```

Parameters

type
The runtime type to unregister.

See Also

Theme Members
Theme Class
MindFusion.Diagramming Namespace

Thickness Structure

Describes the thickness of a rectangular frame.

Syntax

C# Copy Code

```
public struct Thickness
```

Visual Basic Copy Code

```
Public Structure Thickness
```

See Also

Thickness Members
MindFusion.Diagramming Namespace

Thickness Members

Describes the thickness of a rectangular frame.

The following tables list the members exposed by the Thickness type.

Public Constructors

	Name	Description
	Thickness	Overloaded.

Public Properties

	Name	Description
	Bottom	Gets or sets the width of the bottom side of the frame.
	Height	Returns the sum of Top and Bottom margins.
	Left	Gets or sets the width of the left side of the frame.
	Right	Gets or sets the width of the right side of the frame.
	Top	Gets or sets the width of the top side of the frame.
	Width	Returns the sum of Left and Right margins.

See Also

Thickness Structure
MindFusion.Diagramming Namespace

Thickness Constructor

Overload List

Name	Description
Thickness (Double)	Initializes a new instance of the Thickness structure.
Thickness (Double, Double, Double, Double)	Initializes a new instance of the Thickness structure.

See Also

Thickness Structure
MindFusion.Diagramming Namespace

Thickness Constructor (Double)

Initializes a new instance of the Thickness structure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Thickness (
    double uniformLength
)
```

Visual Basic Copy Code

```
Public New ( _
    uniformLength As Double _
)
```

Parameters

uniformLength

See Also

Thickness Constructor Overload List
Thickness Structure
MindFusion.Diagramming Namespace

Thickness Constructor (Double, Double, Double, Double)

Initializes a new instance of the Thickness structure.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Thickness (
    double left,
    double top,
    double right,
    double bottom
)
```

Visual Basic

```
Public New ( _
    left As Double, _
    top As Double, _
    right As Double, _
    bottom As Double _
)
```

Parameters

left

top

right

bottom

See Also

Thickness Constructor Overload List
Thickness Structure
MindFusion.Diagramming Namespace

Thickness Properties

Thickness.Bottom Property

Gets or sets the width of the bottom side of the frame.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double Bottom { get; set; }
```

Visual Basic

```
Public Property Bottom As Double
```

Property Value

See Also

Thickness Members
Thickness Structure
MindFusion.Diagramming Namespace

Thickness.Height Property

Returns the sum of Top and Bottom margins.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double Height { get; }
```

Visual Basic

```
Public ReadOnly Property Height As Double
```

Property Value

See Also

Thickness Members
Thickness Structure
MindFusion.Diagramming Namespace

Thickness.Left Property

Gets or sets the width of the left side of the frame.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Left { get; set; }
```

Visual Basic Copy Code

```
Public Property Left As Double
```

Property Value

See Also

Thickness Members
Thickness Structure
MindFusion.Diagramming Namespace

Thickness.Right Property

Gets or sets the width of the right side of the frame.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Right { get; set; }
```

Visual Basic Copy Code

```
Public Property Right As Double
```

Property Value

See Also

Thickness Members
Thickness Structure
MindFusion.Diagramming Namespace

Thickness.Top Property

Gets or sets the width of the top side of the frame.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Top { get; set; }
```

Visual Basic Copy Code

```
Public Property Top As Double
```

Property Value

See Also

Thickness Members
Thickness Structure
MindFusion.Diagramming Namespace

Thickness.Width Property

Returns the sum of Left and Right margins.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Width { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Width As Double
```

Property Value

See Also

Thickness Members
Thickness Structure
MindFusion.Diagramming Namespace

ThicknessConverter Class

The **TypeConverter** of the Thickness structure.

Syntax

C# Copy Code

```
public class ThicknessConverter : TypeConverter
```

Visual Basic Copy Code

```
Public Class ThicknessConverter
    Inherits TypeConverter
```

Inheritance Hierarchy

System.Object
  System.ComponentModel.TypeConverter
    MindFusion.Diagramming.ThicknessConverter

See Also

ThicknessConverter Members
MindFusion.Diagramming Namespace

ThicknessConverter Members

The **TypeConverter** of the Thickness structure.

The following tables list the members exposed by the ThicknessConverter type.

Public Constructors

	Name	Description
	ThicknessConverter	Initializes a new instance of the **ThicknessConverter** class.

Public Methods

	Name	Description
	CanConvertFrom	**TypeConverter.CanConvertFrom** override.
	CanConvertTo	**TypeConverter.CanConvertTo** override.
	ConvertFrom	**TypeConverter.ConvertFrom** override.
	ConvertTo	**TypeConverter.ConvertTo** override.

See Also

ThicknessConverter Class
MindFusion.Diagramming Namespace

ThicknessConverter Constructor

Initializes a new instance of the ThicknessConverter class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ThicknessConverter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ThicknessConverter Class
MindFusion.Diagramming Namespace

ThicknessConverter Methods

ThicknessConverter.CanConvertFrom Method

TypeConverter.CanConvertFrom override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool CanConvertFrom (
    ITypeDescriptorContext context,
    Type sourceType
)
```

Visual Basic Copy Code

```
Public Overrides Function CanConvertFrom( _
    context As ITypeDescriptorContext, _
    sourceType As Type _
) As Boolean
```

Parameters

context
An **ITypeDescriptorContext** that provides a format context.

sourceType
A **Type** that represents the type you want to convert from.

Return Value

See Also

ThicknessConverter Members
ThicknessConverter Class
MindFusion.Diagramming Namespace

ThicknessConverter.CanConvertTo Method

TypeConverter.CanConvertTo override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool CanConvertTo (
    ITypeDescriptorContext context,
    Type destinationType
)
```

Visual Basic Copy Code

```
Public Overrides Function CanConvertTo( _
    context As ITypeDescriptorContext, _
    destinationType As Type _
) As Boolean
```

Parameters

context
An **ITypeDescriptorContext** that provides a format context.

destinationType
A **Type** that represents the type you want to convert to.

Return Value

See Also

ThicknessConverter Members
ThicknessConverter Class
MindFusion.Diagramming Namespace

ThicknessConverter.ConvertFrom Method

TypeConverter.ConvertFrom override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Object ConvertFrom (
    ITypeDescriptorContext context,
    CultureInfo culture,
    Object value
)
```

Visual Basic Copy Code

```
Public Overrides Function ConvertFrom( _
    context As ITypeDescriptorContext, _
    culture As CultureInfo, _
    value As Object _
) As Object
```

Parameters

context
An **ITypeDescriptorContext** that provides a format context.

culture
The **CultureInfo** to use as the current culture.

value
The **Object** to convert.

Return Value

See Also

ThicknessConverter Members
ThicknessConverter Class
MindFusion.Diagramming Namespace

ThicknessConverter.ConvertTo Method

TypeConverter.ConvertTo override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Object ConvertTo (
    ITypeDescriptorContext context,
    CultureInfo culture,
```

```
    Object value,
    Type destinationType
)
```

Visual Basic Copy Code

```
Public Overrides Function ConvertTo( _
    context As ITypeDescriptorContext, _
    culture As CultureInfo, _
    value As Object, _
    destinationType As Type _
) As Object
```

Parameters

context

An **ITypeDescriptorContext** that provides a format context.

culture

A **CultureInfo**. If **null** is passed, the current culture is assumed.

value

The **Object** to convert.

destinationType

An **Object** that represents the converted value.

Return Value

See Also

ThicknessConverter Members
ThicknessConverter Class
MindFusion.Diagramming Namespace

TimerEventArgs Class

Contains the arguments passed to timer-related event handlers.

Syntax

C# Copy Code

```
public class TimerEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class TimerEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.TimerEventArgs

See Also

TimerEventArgs Members
MindFusion.Diagramming Namespace

TimerEventArgs Members

Contains the arguments passed to timer-related event handlers.

The following tables list the members exposed by the TimerEventArgs type.

Public Constructors

	Name	Description
	TimerEventArgs	Overloaded.

Public Properties

	Name	Description
	Handler	Gets the tick handler of the timer when a new timer is requested.
	Interval	Gets the timer interval when a new timer is requested.
	IsHandled	Gets or sets a value indicating whether the event is handled.
	IsStart	Gets a value indicating whether a new timer is being created or an existing timer is being stopped.
	Timer	Gets or sets the timer related to the operation.

See Also

TimerEventArgs Class
MindFusion.Diagramming Namespace

TimerEventArgs Constructor

Overload List

Name	Description
TimerEventArgs (Int32, ElapsedEventHandler)	Initializes a new instance of the TimerEventArgs class.
TimerEventArgs (Object)	Initializes a new instance of the TimerEventArgs class.

See Also

TimerEventArgs Class
MindFusion.Diagramming Namespace

TimerEventArgs Constructor (Int32, ElapsedEventHandler)

Initializes a new instance of the TimerEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TimerEventArgs (
    int interval,
    ElapsedEventHandler handler
)
```

Visual Basic Copy Code

```
Public New ( _
    interval As Integer, _
    handler As ElapsedEventHandler _
)
```

Parameters

interval

handler

See Also

TimerEventArgs Constructor Overload List
TimerEventArgs Class
MindFusion.Diagramming Namespace

TimerEventArgs Constructor (Object)

Initializes a new instance of the TimerEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TimerEventArgs (
    Object timer
)
```

Visual Basic Copy Code

```
Public New ( _
    timer As Object _
)
```

Parameters

timer

See Also

TimerEventArgs Constructor Overload List
TimerEventArgs Class
MindFusion.Diagramming Namespace

TimerEventArgs Properties

TimerEventArgs.Handler Property

Gets the tick handler of the timer when a new timer is requested.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ElapsedEventHandler Handler { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Handler As ElapsedEventHandler
```

Property Value

See Also

TimerEventArgs Members
TimerEventArgs Class
MindFusion.Diagramming Namespace

TimerEventArgs.Interval Property

Gets the timer interval when a new timer is requested.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int Interval { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Interval As Integer
```

Property Value

See Also

TimerEventArgs Members
TimerEventArgs Class
MindFusion.Diagramming Namespace

TimerEventArgs.IsHandled Property

Gets or sets a value indicating whether the event is handled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsHandled { get; set; }
```

Visual Basic Copy Code

```
Public Property IsHandled As Boolean
```

Property Value

See Also

TimerEventArgs Members
TimerEventArgs Class
MindFusion.Diagramming Namespace

TimerEventArgs.IsStart Property

Gets a value indicating whether a new timer is being created or an existing timer is being stopped.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IsStart { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property IsStart As Boolean
```

Property Value

See Also

TimerEventArgs Members
TimerEventArgs Class
MindFusion.Diagramming Namespace

TimerEventArgs.Timer Property

Gets or sets the timer related to the operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Timer { get; set; }
```

Visual Basic Copy Code

```
Public Property Timer As Object
```

Property Value

See Also

TimerEventArgs Members
TimerEventArgs Class
MindFusion.Diagramming Namespace

TreeEnumerator Class

Enables an easy standard method to iterate over the elements of a tree-like structure of objects.

Syntax

C# Copy Code

```
public class TreeEnumerator : IEnumerator
```

Visual Basic Copy Code

```
Public Class TreeEnumerator
    Implements IEnumerator
```

Remarks

The **TreeEnumerator** class implements the standard .NET **IEnumerator** interface. It provides an easy method to iterate over the elements of a tree-like structure of diagram items.

Example

The following example demonstrates how to use the **TreeEnumerator**:

C# Copy Code

```
TreeEnumerator te = new TreeEnumerator(
    rootObj, false, false);

while (te.MoveNext())
{
    ShapeNode node = (ShapeNode)te.Current;
    node.Brush = new SolidBrush(Color.Blue);
}
```

Visual Basic Copy Code

```
Dim te As MindFusion.Diagramming.TreeEnumerator = New MindFusion.Diagramming.TreeEnumerator( _
    rootObj, False, False)

While te.MoveNext()
    Dim node As ShapeNode = CType(te.Current, ShapeNode)
    Node.Brush = New SolidBrush(Color.Blue)
End While
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.TreeEnumerator

See Also

TreeEnumerator Members
MindFusion.Diagramming Namespace

TreeEnumerator Members

Enables an easy standard method to iterate over the elements of a tree-like structure of objects.

The following tables list the members exposed by the TreeEnumerator type.

Public Constructors

	Name	Description
	TreeEnumerator	Initializes a new instance of the **TreeEnumerator** class.

Public Properties

	Name	Description
	Current	Gets the current element of the iteration.

Public Methods

	Name	Description
	MoveNext	Advances the enumerator to the next element of the tree.
	Reset	Sets the enumerator to its initial position, which is before the first element in the collection.

See Also

TreeEnumerator Class
MindFusion.Diagramming Namespace

TreeEnumerator Constructor

Initializes a new instance of the TreeEnumerator class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeEnumerator (
    DiagramNode root,
    bool enumLinks,
    bool strict
)
```

Visual Basic Copy Code

```
Public New ( _
    root As DiagramNode, _
    enumLinks As Boolean, _
    strict As Boolean _
)
```

Parameters

root
The root of the tree or subtree to be enumerated.

enumLinks
Specifies whether to enumerate links.

strict
Specifies whether to check strictly if the passed structure is a tree. If this option is enabled, NotATreeException will be thrown if a cycle or multiple links to a node are encountered. If the option is disabled, the enumerator will ignore any cycles or multiple links found and continue the iteration.

See Also

TreeEnumerator Class
MindFusion.Diagramming Namespace

TreeEnumerator Methods

TreeEnumerator.MoveNext Method

Advances the enumerator to the next element of the tree.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool MoveNext ()
```

Visual Basic Copy Code

```
Public Function MoveNext () As Boolean
```

Return Value

true if the enumerator has successfully passed to the next DiagramItem in the collection; **false** if there are no more elements available to enumerate.

Remarks

Initially an enumerator is positioned before the first element of the tree (the root), and the first call to **MoveNext** moves the enumerator over that first element.

See Also

TreeEnumerator Members
TreeEnumerator Class
MindFusion.Diagramming Namespace

TreeEnumerator.Reset Method

Sets the enumerator to its initial position, which is before the first element in the collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Reset ()
```

Visual Basic Copy Code

```
Public Sub Reset ()
```

See Also

TreeEnumerator Members
TreeEnumerator Class
MindFusion.Diagramming Namespace

TreeEnumerator Properties

TreeEnumerator.Current Property

Gets the current element of the iteration.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Current { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Current As Object
```

Property Value

The DiagramItem that is current in the enumeration.

Remarks

If the *enumLinks* constructor argument is set to **false**, only nodes are enumerated. After an enumerator is created or after a Reset, MoveNext must be called to advance the enumerator to the first element before reading the value of **Current**; otherwise, **Current** is undefined.

See Also

TreeEnumerator Members
TreeEnumerator Class

MindFusion.Diagramming Namespace

TreeItemEventArgs Class

Contains the arguments passed to handlers of TreeViewItem-related events.

Syntax

C# Copy Code

```
public class TreeItemEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class TreeItemEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.TreeItemEventArgs
      MindFusion.Diagramming.TreeItemValidationEventArgs

See Also

TreeItemEventArgs Members
MindFusion.Diagramming Namespace

TreeItemEventArgs Members

Contains the arguments passed to handlers of TreeViewItem-related events.

The following tables list the members exposed by the TreeItemEventArgs type.

Public Constructors

	Name	Description
	TreeItemEventArgs	Initializes a new instance of the **TreeItemEventArgs** class.

Public Properties

	Name	Description
	Item	Get the item for which the event is raised.
	MouseButton	Gets which mouse button has been pressed.
	MousePosition	Gets the current mouse pointer position.

See Also

TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemEventArgs Constructor

Initializes a new instance of the TreeItemEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeItemEventArgs (
    TreeViewItem item,
    MouseButton mouseButton,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public New ( _
    item As TreeViewItem, _
    mouseButton As MouseButton, _
    mousePosition As Point _
)
```

Parameters

item
The TreeViewItem related to the event.

mouseButton
The mouse button related to the event.

mousePosition
The position of the mouse at the time the event was raised.

See Also

TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemEventArgs Properties

TreeItemEventArgs.Item Property

Get the item for which the event is raised.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewItem Item { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Item As TreeViewItem
```

Property Value

See Also

TreeItemEventArgs Members
TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemEventArgs.MouseButton Property

Gets which mouse button has been pressed.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MouseButton MouseButton { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MouseButton As MouseButton
```

Property Value

See Also

TreeItemEventArgs Members
TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemEventArgs.MousePosition Property

Gets the current mouse pointer position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point MousePosition { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MousePosition As Point
```

Property Value

See Also

TreeItemEventArgs Members
TreeItemEventArgs Class
MindFusion.Diagramming Namespace

TreeItemValidationEventArgs Class

Contains the arguments passed to event handlers, such as TreeItemTextEditing.

Syntax

C# Copy Code

```
public class TreeItemValidationEventArgs : TreeItemEventArgs
```

Visual Basic Copy Code

```
Public Class TreeItemValidationEventArgs
    Inherits TreeItemEventArgs
```

Inheritance Hierarchy

System.Object
System.EventArgs
MindFusion.Diagramming.TreeItemEventArgs
MindFusion.Diagramming.TreeItemValidationEventArgs

See Also

TreeItemValidationEventArgs Members
MindFusion.Diagramming Namespace

TreeItemValidationEventArgs Members

Contains the arguments passed to event handlers, such as TreeItemTextEditing.

The following tables list the members exposed by the TreeItemValidationEventArgs type.

Public Constructors

	Name	Description
	TreeItemValidationEventArgs	Initializes a new instance of the **TreeItemValidationEventArgs** class.

Public Properties

	Name	Description
	Cancel	Gets or sets a value indicating whether to prevent the user from completing the operation.
	Item	Get the item for which the event is raised. (Inherited from TreeItemEventArgs.)
	MouseButton	Gets which mouse button has been pressed. (Inherited from TreeItemEventArgs.)
	MousePosition	Gets the current mouse pointer position. (Inherited from TreeItemEventArgs.)

See Also

TreeItemValidationEventArgs Class
MindFusion.Diagramming Namespace

TreeItemValidationEventArgs Constructor

Initializes a new instance of the TreeItemValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeItemValidationEventArgs (
    TreeViewItem item,
    MouseButton mouseButton,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Public New ( _
    item As TreeViewItem, _
    mouseButton As MouseButton, _
    mousePosition As Point _
)
```

Parameters

item
The TreeViewItem related to the event.

mouseButton
The mouse button related to the event.

mousePosition
The position of the mouse at the time the event was raised.

See Also

TreeItemValidationEventArgs Class
MindFusion.Diagramming Namespace

TreeItemValidationEventArgs Properties

TreeItemValidationEventArgs.Cancel Property

Gets or sets a value indicating whether to prevent the user from completing the operation.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Cancel { get; set; }
```

Visual Basic Copy Code

```
Public Property Cancel As Boolean
```

Property Value

See Also

TreeItemValidationEventArgs Members
TreeItemValidationEventArgs Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint Class

Represents a connection between a link and a TreeViewNode.

Syntax

C# Copy Code

```
public class TreeViewConnectionPoint : ConnectionPoint
```

Visual Basic Copy Code

```
Public Class TreeViewConnectionPoint
    Inherits ConnectionPoint
```

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.ConnectionPoint
MindFusion.Diagramming.TreeViewConnectionPoint

See Also

TreeViewConnectionPoint Members
MindFusion.Diagramming Namespace

TreeViewConnectionPoint Members

Represents a connection between a link and a TreeViewNode.

The following tables list the members exposed by the TreeViewConnectionPoint type.

Public Constructors

	Name	Description
	TreeViewConnectionPoint	Overloaded.

Public Properties

	Name	Description
	Item	Gets the item to which the link connects.
	RelativePosition	Gets or sets the position of this connection point, relative to the associated node's bounding rectangle, expressed as percentage. (Inherited from ConnectionPoint.)
	Row	ConnectionPoint.Row override.

Protected Properties

	Name	Description
	Incoming	Gets a flag indicating whether this connection point is at the origin or the destination of the associated link. (Inherited from ConnectionPoint.)
	Link	Gets the link associated with this connection point. (Inherited from ConnectionPoint.)
	Node	Gets the node associated with this connection point. (Inherited from ConnectionPoint.)

Public Methods

	Name	Description
	AddLinkToNode	ConnectionPoint.AddLinkToNode override.
	CalculateIntersections	Determines whether the end of a link should be aligned to the point where the node contour is intersected by the line connecting the centers of the connected nodes. (Inherited from ConnectionPoint.)
	CanMoveLink	Checks whether the specified point, expressed in document coordinates, is a valid location for this connection point. (Inherited from ConnectionPoint.)
	GetAnchorPos	Overloaded. (Inherited from ConnectionPoint.)
	GetEndPoint	ConnectionPoint.GetEndPoint override.
	GetInitialPoint	ConnectionPoint.GetInitialPoint override.

	GetIntersection	ConnectionPoint.GetIntersection override.
	GetNodeRect	Returns the bounding rectangle of the node associated with this connection point. (Inherited from ConnectionPoint.)
	LinkChanges	ConnectionPoint.LinkChanges override.
	NearestAnchorPoint	Returns the anchor point closest to the specified point from the node's associated anchor pattern. (Inherited from ConnectionPoint.)
	NodesIntersect	Checks whether the node associated with the current connection point and the node associated with the specified connection point intersect. (Inherited from ConnectionPoint.)
	RemoveLinkFromNode	ConnectionPoint.RemoveLinkFromNode override.
	SameNode	Checks whether the current connection point and the specified connection point are associated with the same node. (Inherited from ConnectionPoint.)
	SaveEndRelative	ConnectionPoint.SaveEndRelative override.

Protected Methods

	Name	Description
	GetNearestBorderPoint	ConnectionPoint.GetNearestBorderPoint override.
	UpdateFromPoint	Updates the connection stated according to the specified point, without changing the related node.

See Also

TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint Constructor

Overload List

Name	Description
TreeViewConnectionPoint ()	Initializes a new instance of the TreeViewConnectionPoint class.
TreeViewConnectionPoint (TreeViewNode, DiagramLink, Boolean, TreeViewItem)	Initializes a new instance of the **TreeViewConnectionPoint** class.

See Also

TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint Constructor ()

Initializes a new instance of the TreeViewConnectionPoint class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewConnectionPoint ()
```

Visual Basic

```
Public Sub New ()
```

Remarks

This constructor is used only during deserialization.

See Also

TreeViewConnectionPoint Constructor Overload List
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint Constructor (TreeViewNode, DiagramLink, Boolean, TreeViewItem)

Initializes a new instance of the TreeViewConnectionPoint class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public TreeViewConnectionPoint (
    TreeViewNode node,
    DiagramLink link,
    bool incoming,
    TreeViewItem item
)
```

Visual Basic

```
Public New ( _
    node As TreeViewNode, _
    link As DiagramLink, _
    incoming As Boolean, _
    item As TreeViewItem _
)
```

Parameters

node
A TreeViewNode instance.

link
A DiagramLink instance.

incoming
Specifies which end of the link is represented by this TreeViewConnectionPoint.

item
The TreeViewItem to which the link connects.

See Also

TreeViewConnectionPoint Constructor Overload List
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint Methods

TreeViewConnectionPoint.AddLinkToNode Method

ConnectionPoint.AddLinkToNode override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void AddLinkToNode ()
```

Visual Basic Copy Code

```
Public Overrides Sub AddLinkToNode ()
```

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.GetEndPoint Method

ConnectionPoint.GetEndPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Point GetEndPoint ()
```

Visual Basic Copy Code

```
Public Overrides Function GetEndPoint () As Point
```

Return Value

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.GetInitialPoint Method

ConnectionPoint.GetInitialPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Point GetInitialPoint ()
```

Visual Basic Copy Code

```
Public Overrides Function GetInitialPoint () As Point
```

Return Value

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.GetIntersection Method

ConnectionPoint.GetIntersection override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Point GetIntersection (
    Point pt1,
    Point pt2
)
```

Visual Basic Copy Code

```
Public Overrides Function GetIntersection( _
    pt1 As Point, _
    pt2 As Point _
) As Point
```

Parameters

pt1

pt2

Return Value

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.GetNearestBorderPoint Method

ConnectionPoint.GetNearestBorderPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override Point GetNearestBorderPoint (
    Point point
)
```

Visual Basic Copy Code

```
Protected Overrides Function GetNearestBorderPoint( _
    point As Point _
) As Point
```

Parameters

point

Return Value

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.LinkChanges Method

ConnectionPoint.LinkChanges override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool LinkChanges (
    DiagramItem item,
    Point point
)
```

Visual Basic Copy Code

```
Public Overrides Function LinkChanges( _
    item As DiagramItem, _
    point As Point _
) As Boolean
```

Parameters

item

point

Return Value

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.RemoveLinkFromNode Method

ConnectionPoint.RemoveLinkFromNode override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void RemoveLinkFromNode ()
```

Visual Basic Copy Code

```
Public Overrides Sub RemoveLinkFromNode ()
```

See Also

TreeViewConnectionPoint.SaveEndRelative Method

ConnectionPoint.SaveEndRelative override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void SaveEndRelative ()
```

Visual Basic Copy Code

```
Public Overrides Sub SaveEndRelative ()
```

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.UpdateFromPoint Method

Updates the connection stated according to the specified point, without changing the related node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void UpdateFromPoint (
    Point point
)
```

Visual Basic Copy Code

```
Protected Overrides Sub UpdateFromPoint( _
    point As Point _
)
```

Parameters

point

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint Properties

TreeViewConnectionPoint.Item Property

Gets the item to which the link connects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewItem Item { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Item As TreeViewItem
```

Property Value

A TreeViewItem instance specifying the connection's target item.

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionPoint.Row Property

ConnectionPoint.Row override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override int Row { get; set; }
```

Visual Basic Copy Code

```
Public Overrides Property Row As Integer
```

Property Value

See Also

TreeViewConnectionPoint Members
TreeViewConnectionPoint Class
MindFusion.Diagramming Namespace

TreeViewConnectionStyle Enumeration

Specifies the possible values for the ConnectionStyle property of TreeViewNode objects.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum TreeViewConnectionStyle
```

Visual Basic	Copy Code
`Public Enum TreeViewConnectionStyle`	

Members

	Member name	Description
	Node	Users are allowed to create relations to the TreeViewNode as an integral entity.
	Items	Users are allowed to create relations to the items of a **TreeViewNode**.

See Also

MindFusion.Diagramming Namespace

TreeViewItem Class

Represents a constituent item of the hierarchy displayed inside a TreeViewNode.

Syntax

C#	Copy Code
`public class TreeViewItem : INotifyPropertyChanged, InplaceEditable`	

Visual Basic	Copy Code
`Public Class TreeViewItem` `    Implements INotifyPropertyChanged, InplaceEditable`	

Remarks

The items of a TreeViewNode can be traversed recursively starting from the node's RootItems. Given an item, you can find its parent and child items by means of the Parent and Children properties.

If the ConnectionStyle of a node is set to Items, links connect to specific items of the node. Such links can be accessed by means of the IncomingLinks and OutgoingLinks collections.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.TreeViewItem

See Also

TreeViewItem Members
MindFusion.Diagramming Namespace

TreeViewItem Members

Represents a constituent item of the hierarchy displayed inside a TreeViewNode.

The following tables list the members exposed by the TreeViewItem type.

Public Constructors

	Name	Description
	TreeViewItem	Overloaded. Initializes a new instance of the **TreeViewItem** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to paint item's background.
	Children	Gets a list of this item's children in the hierarchy.
	Expanded	Gets or sets whether this item is expanded to show its children.
	Height	Gets or sets the height of this item.
	Image	Gets or sets the image that is displayed in this treeview item.
	ImageUrl	Gets or sets a URL used to load an image into this item.
	IncomingLinks	Gets a collection of all links that point to this item.
	Label	Gets or sets the label of this item.
	LabelBrush	Gets or sets the brush used to draw the label of this item.
	LabelFont	Gets or sets the label font of this item.
	OutgoingLinks	Gets a collection of all links that go out of this item.
	Parent	Gets the parent item of this item.
	Tag	Gets or sets custom data associated with this item.

Public Events

	Name	Description
	PropertyChanged	Implements INotifyPropertyChanged.PropertyChanged.

See Also

TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem Constructor

Initializes a new instance of the TreeViewItem class.

Overload List

Name	Description
TreeViewItem ()	Initializes a new instance of the **TreeViewItem** class.
TreeViewItem (String)	Initializes a new instance of the **TreeViewItem** class.
TreeViewItem (TreeViewItem)	Initializes a new instance of the **TreeViewItem** class using the specified item as a prototype.

See Also

TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem Constructor ()

Initializes a new instance of the TreeViewItem class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewItem ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

TreeViewItem Constructor Overload List
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem Constructor (String)

Initializes a new instance of the TreeViewItem class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewItem (
    string label
)
```

Visual Basic Copy Code

```
Public New ( _
    label As String _
)
```

Parameters

label
The text initially displayed by the item.

See Also

TreeViewItem Constructor Overload List
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem Constructor (TreeViewItem)

Initializes a new instance of the TreeViewItem class using the specified item as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewItem (
    TreeViewItem prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As TreeViewItem _
)
```

Parameters

prototype
The treeview item to use as a prototype. The properties of the new treeview are initialized from the property values of prototype.

Remarks

This constructor copies the TreeViewItem property values from the specified prototype item.

See Also

TreeViewItem Constructor Overload List
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem Properties

TreeViewItem.Brush Property

Gets or sets the brush used to paint item's background.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush Brush { get; set; }
```

Visual Basic Copy Code

```
Public Property Brush As Brush
```

Property Value

An instance of Brush -derived class. The default value is **null**.

Remarks

If this property is not set, item's background is painted using the parent node's Brush.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.Children Property

Gets a list of this item's children in the hierarchy.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IList<TreeViewItem> Children { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Children As IList(Of TreeViewItem)
```

Property Value

A list of child TreeViewItem objects.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace
Parent Property

TreeViewItem.Expanded Property

Gets or sets whether this item is expanded to show its children.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Expanded { get; set; }
```

Visual Basic Copy Code

```
Public Property Expanded As Boolean
```

Property Value

true to show the children of this item, otherwise **false**; the default value is **true**.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.Height Property

Gets or sets the height of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Height { get; set; }
```

Visual Basic Copy Code

```
Public Property Height As Double
```

Property Value

A **double** value specifying the item's height. The default value is 5.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.Image Property

Gets or sets the image that is displayed in this treeview item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage Image { get; set; }
```

Visual Basic Copy Code

```
Public Property Image As IImage
```

Property Value

A **Microsoft.Maui.Graphics.IImage** instance.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.ImageUrl Property

Gets or sets a URL used to load an image into this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ImageUrl { get; set; }
```

Visual Basic Copy Code

```
Public Property ImageUrl As String
```

Property Value

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.IncomingLinks Property

Gets a collection of all links that point to this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection IncomingLinks { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property IncomingLinks As DiagramLinkCollection
```

Property Value

An instance of the DiagramLinkCollection class.

Remarks

The property returns a collection of all links that point to this item. Use it to access the links or get their number.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.Label Property

Gets or sets the label of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Label { get; set; }
```

Visual Basic Copy Code

```
Public Property Label As String
```

Property Value

A string specifying the item's label.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace
LabelBrush Property
LabelFont Property
TreeViewNode.ResizeToFitText Method

TreeViewItem.LabelBrush Property

Gets or sets the brush used to draw the label of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush LabelBrush { get; set; }
```

Visual Basic Copy Code

```
Public Property LabelBrush As Brush
```

Property Value

An instance of a MindFusion.Drawing.Brush-derived class. The default value is **null** (**Nothing** in Visual Basic).

Remarks

It this property is set to **null**, the label is rendered using the TextBrush value of the parent node.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace
Label Property
LabelFont Property

TreeViewItem.LabelFont Property

Gets or sets the label font of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Font LabelFont { get; set; }
```

Visual Basic Copy Code

```
Public Property LabelFont As Font
```

Property Value

An instance of the Font class. The default value is **null**.

Remarks

It this property is set to **null**, the label is rendered using the Font of the parent node.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace
Label Property
LabelBrush Property

TreeViewItem.OutgoingLinks Property

Gets a collection of all links that go out of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramLinkCollection OutgoingLinks { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property OutgoingLinks As DiagramLinkCollection
```

Property Value

An instance of the DiagramLinkCollection class.

Remarks

The property returns a collection of all links that start from this item. Use it to access the links or get their number.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem.Parent Property

Gets the parent item of this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewItem Parent { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Parent As TreeViewItem
```

Property Value

A TreeViewItem instance.

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace
Children Property

TreeViewItem.Tag Property

Gets or sets custom data associated with this item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object Tag { get; set; }
```

Visual Basic Copy Code

```
Public Property Tag As Object
```

Property Value

An instance of any class derived from **System.Object**. The default value is **null** (**Nothing** in Visual Basic).

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewItem Events

TreeViewItem.PropertyChanged Event

Implements INotifyPropertyChanged.PropertyChanged.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public event PropertyChangedEventHandler PropertyChanged
```

Visual Basic Copy Code

```
Public Event PropertyChanged As PropertyChangedEventHandler
```

See Also

TreeViewItem Members
TreeViewItem Class
MindFusion.Diagramming Namespace

TreeViewNode Class

Represents nodes that can draw hierarchical data.

Syntax

C# Copy Code

```
public class TreeViewNode : DiagramNode, Scrollable
```

Visual Basic Copy Code

```
Public Class TreeViewNode
    Inherits DiagramNode
    Implements Scrollable
```

Remarks

The root items displayed in the node can be accessed through the RootItems property. New items can be added to and removed from the returned collection. The root items can have child items on their own and so on. For more information, refer to the TreeViewItem class.

The text and image displayed in the node caption can be customized through the Caption and CaptionImage properties respectively. The scroll position can be manipulated through ScrollPosition.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.TreeViewNode

See Also

TreeViewNode Members
MindFusion.Diagramming Namespace

TreeViewNode Members

Represents nodes that can draw hierarchical data.

The following tables list the members exposed by the TreeViewNode type.

Public Constructors

	Name	Description
	TreeViewNode	Overloaded. Initializes a new instance of the **TreeViewNode** class.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Caption	Gets or sets the text displayed as caption of the table.
	CaptionHeight	Gets or sets the height of the node's caption bar.
	CaptionImage	Gets or sets the image that is displayed in the caption.
	ConnectionStyle	Gets or sets a value indicating whether drawing a link from/to this node should connect an item or the whole node as an integral entity.
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)

	EnableStyledText	Gets or sets a value indicating whether styled text rendering is enabled.
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the node's caption.
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RootItems	Gets a list of the root TreeViewItem objects displayed in this node.
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	ScrollPosition	Gets or sets the current scroll position.
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)

	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Shape	Gets or sets the shape of the table's outline.
	ShowLines	Gets or sets whether to display lines connecting child items to parent items.
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of texts displayed inside this shape node.
	TextPadding	Gets or sets the spacing between the item boundaries and its text.
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	DiagramNode.HasItems override.
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	DiagramNode.AcceptLinks override.
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)

	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	DiagramNode.CreateConnectionPoint override.
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Draws the node on the specified IGraphics surface.
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified **IGraphics** surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetFlatItemList	Gets a list with all items of this node.
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetItemRect	Gets the location and size of the specified TreeViewItem.
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)

	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTest	Returns the type of graphics element at the specified coordinates.
	HitTestExpander	Gets whether there is a collapse/expand button located at the specified position.
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	ItemFromPoint	Returns the item at the specified location.
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	DiagramItem.OnLoad override.
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveItem	Removes the specified item from the node.
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	ResizeToFitText	Resizes the tree view so that it is large enough to fit the text of its items.
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)

	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override.
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllIncomingLinks	DiagramNode.GetAllIncomingLinks override.
	GetAllOutgoingLinks	DiagramNode.GetAllOutgoingLinks override.
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFromJson	Loads the node from a JSON object.
	LoadFromXml	Loads the item content from an XML element.

	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	DiagramNode.OnClick override.
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	Called when the Diagram's MeasureUnit changes.
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	DiagramItem.OnStyleChanged override.
	OnTextChanged	DiagramItem.OnTextChanged override.
	OnTextFormatChanged	DiagramItem.OnTextFormatChanged override.
	OnUpdateBounds	DiagramNode.OnUpdateBounds override.
	OnZoomChanged	Called when the Diagram's **ZoomFactor** changes.
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramNode.RestoreProperties override.
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramNode.SaveProperties override.
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveToJson	Saves the node as a JSON object.
	SaveToXml	Saves the item content into an XML element.
	ScaleElements	DiagramNode.ScaleElements override.
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it.
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)

	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode Constructor

Initializes a new instance of the TreeViewNode class.

Overload List

Name	Description
TreeViewNode ()	Initializes a new instance of the **TreeViewNode** class.
TreeViewNode (Diagram)	Initializes a new instance of the **TreeViewNode** class.
TreeViewNode (TreeViewNode)	Initializes a new instance of the **TreeViewNode** class using the specified node as a prototype.

See Also

TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode Constructor ()

Initializes a new instance of the TreeViewNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewNode ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

TreeViewNode Constructor Overload List
TreeViewNode Class

TreeViewNode Constructor (Diagram)

Initializes a new instance of the TreeViewNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewNode (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram _
)
```

Parameters

diagram
The Diagram from which to obtain default values for the node properties.

Remarks

This constructor initializes the TreeViewNode properties with the values specified in the respective default-value properties of the Diagram object.

See Also

TreeViewNode Constructor Overload List
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode Constructor (TreeViewNode)

Initializes a new instance of the TreeViewNode class using the specified node as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewNode (
    TreeViewNode prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As TreeViewNode _
)
```

Parameters

prototype
The treeview node to use as a prototype. The properties of the new treeview are initialized from the property values of prototype.

Remarks

This constructor copies the TreeViewNode property values from the specified prototype node.

See Also

TreeViewNode Constructor Overload List
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode Methods

TreeViewNode.AcceptLinks Method

DiagramNode.AcceptLinks override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool AcceptLinks (
    bool outgoing
)
```

Visual Basic Copy Code

```
Public Overrides Function AcceptLinks( _
    outgoing As Boolean _
) As Boolean
```

Parameters

outgoing
true to check for outgoing links; otherwise, **false**.

Return Value

true if the node accepts connections of the specified type; otherwise, **false**.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.CreateConnectionPoint Method

DiagramNode.CreateConnectionPoint override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override ConnectionPoint CreateConnectionPoint (
    DiagramLink link,
    Point point,
    bool incoming
)
```

Visual Basic Copy Code

```
Public Overrides Function CreateConnectionPoint( _
    link As DiagramLink, _
```

```
    point As Point, _
    incoming As Boolean _
) As ConnectionPoint
```

Parameters

link
A DiagramLink that should be connected to this node.

point
The point at which the link should be connected.

incoming
true if link is an incoming link; otherwise, **false**.

Return Value

A ConnectionPoint instance representing the connection.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.CreateProperties Method

DiagramItem.CreateProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override DiagramItemProperties CreateProperties ()
```

Visual Basic Copy Code

```
Protected Overrides Function CreateProperties () As DiagramItemProperties
```

Return Value

A DiagramItemProperties-derived object containing a snapshot of the item's properties.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.DrawLocal Method

Draws the node on the specified IGraphics surface.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void DrawLocal (
    IGraphics graphics,
    RenderOptions options
)
```

Visual Basic Copy Code

```
Public Overrides Sub DrawLocal( _
    graphics As IGraphics, _
    options As RenderOptions _
)
```

Parameters

graphics
An IGraphics object where the node should be rendered.

options
Provides options for the drawing operation.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.GetAllIncomingLinks Method

DiagramNode.GetAllIncomingLinks override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void GetAllIncomingLinks (
    DiagramLinkCollection links
)
```

Visual Basic Copy Code

```
Protected Overrides Sub GetAllIncomingLinks( _
    links As DiagramLinkCollection _
)
```

Parameters

links
A DiagramLinkCollection the links should be added to.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.GetAllOutgoingLinks Method

DiagramNode.GetAllOutgoingLinks override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void GetAllOutgoingLinks (
    DiagramLinkCollection links
)
```

Visual Basic Copy Code

```
Protected Overrides Sub GetAllOutgoingLinks( _
    links As DiagramLinkCollection _
)
```

Parameters

links
A DiagramLinkCollection the links should be added to.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.GetFlatItemList Method

Gets a list with all items of this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<TreeViewItem> GetFlatItemList ()
```

Visual Basic Copy Code

```
Public Function GetFlatItemList () As List(Of TreeViewItem)
```

Return Value

A list of TreeViewItem objects.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.GetItemRect Method

Gets the location and size of the specified TreeViewItem.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect GetItemRect (
    TreeViewItem treeItem
)
```

Visual Basic Copy Code

```
Public Function GetItemRect( _
    treeItem As TreeViewItem _
) As Rect
```

Parameters

treeItem
A TreeViewItem instance.

Return Value

A **Microsoft.Maui.Graphics.Rect** specifying the item's position and size.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.HitTest Method

Returns the type of graphics element at the specified coordinates.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public HitTestResult HitTest (
    Point point
)
```

Visual Basic Copy Code

```
Public Function HitTest( _
    point As Point _
) As HitTestResult
```

Parameters

point
A point in the diagram.

Return Value

A member of the HitTestResult enumeration.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.HitTestExpander Method

Gets whether there is a collapse/expand button located at the specified position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool HitTestExpander (
    Point point
)
```

Visual Basic Copy Code

```
Public Function HitTestExpander( _
    point As Point _
) As Boolean
```

Parameters

point
The point to check.

Return Value

true if there is +/- icon at the specified position, otherwise **false**.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.ItemFromPoint Method

Returns the item at the specified location.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewItem ItemFromPoint (
    Point point
)
```

Visual Basic Copy Code

```
Public Function ItemFromPoint( _
    point As Point _
) As TreeViewItem
```

Parameters

point
The point to check.

Return Value

The TreeViewItem instance that contains the specified point.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.LoadFromJson Method

Loads the node from a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
```

```
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject containing the node's data.

context

A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.LoadFromXml Method

Loads the item content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An **XmlElement** containing the item's data.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.OnClick Method

DiagramNode.OnClick override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnClick (
    MouseButton mouseButton,
    Point mousePosition
)
```

Visual Basic Copy Code

```
Protected Overrides Sub OnClick( _
    mouseButton As MouseButton, _
    mousePosition As Point _
)
```

Parameters

mouseButton

mousePosition

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.OnLoad Method

DiagramItem.OnLoad override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override void OnLoad ()
```

Visual Basic Copy Code

```
Public Overrides Sub OnLoad ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.OnMeasureUnitChanged Method

Called when the Diagram's MeasureUnit changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnMeasureUnitChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnMeasureUnitChanged ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.OnStyleChanged Method

DiagramItem.OnStyleChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnStyleChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnStyleChanged ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.OnTextChanged Method

DiagramItem.OnTextChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnTextChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnTextChanged ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.OnTextFormatChanged Method

DiagramItem.OnTextFormatChanged override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnTextFormatChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnTextFormatChanged ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.OnUpdateBounds Method

DiagramNode.OnUpdateBounds override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnUpdateBounds (
    bool sizeChanged
)
```

Visual Basic Copy Code

```
Protected Overrides Sub OnUpdateBounds( _
    sizeChanged As Boolean _
)
```

Parameters

sizeChanged
true if the size has changed; **false** if only the position has changed.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.OnZoomChanged Method

Called when the Diagram's **ZoomFactor** changes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void OnZoomChanged ()
```

Visual Basic Copy Code

```
Protected Overrides Sub OnZoomChanged ()
```

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.RemoveItem Method

Removes the specified item from the node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void RemoveItem (
    TreeViewItem item
)
```

Visual Basic Copy Code

```
Public Sub RemoveItem( _
    item As TreeViewItem _
)
```

Parameters

item
The TreeViewItem to remove.

Remarks

The method removes the specified item and all its direct or indirect children, as well as the links connecting to them.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.ResizeToFitText Method

Resizes the tree view so that it is large enough to fit the text of its items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void ResizeToFitText ()
```

Visual Basic Copy Code

```
Public Sub ResizeToFitText ()
```

Remarks

This method also resizes the node vertically so that its height fits all items.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace
Caption Property
TreeViewItem.Label Property

TreeViewNode.RestoreProperties Method

DiagramNode.RestoreProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void RestoreProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub RestoreProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties-derived object containing a snapshot of the item's properties.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.SaveProperties Method

DiagramNode.SaveProperties override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveProperties (
    DiagramItemProperties props
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveProperties( _
    props As DiagramItemProperties _
)
```

Parameters

props
A DiagramItemProperties-derived object where the item's properties should be saved.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.SaveToJson Method

Saves the node as a JSON object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject instance to contain the serialized data.

context
A JsonPersistContext object providing contextual information about the serialization process and helper serialization methods.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.SaveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An **XmlElement** the item's data should be stored into.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.ScaleElements Method

DiagramNode.ScaleElements override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void ScaleElements (
    double scale
)
```

Visual Basic Copy Code

```
Protected Overrides Sub ScaleElements( _
    scale As Double _
)
```

Parameters

scale
The value by which to multiply node elements' sizes.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.SetReflexive Method

Invoked to let a node set the shape of a reflexive link connected to it.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SetReflexive (
    DiagramLink link
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SetReflexive( _
    link As DiagramLink _
)
```

Parameters

link
Reference to the link.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode Properties

TreeViewNode.Caption Property

Gets or sets the text displayed as caption of the table.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string Caption { get; set; }
```

Visual Basic Copy Code

```
Public Property Caption As String
```

Property Value

A string specifying the caption text.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.CaptionHeight Property

Gets or sets the height of the node's caption bar.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double CaptionHeight { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionHeight As Double
```

Property Value

A **double** value specifying the caption bar height. The default value is 8.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.CaptionImage Property

Gets or sets the image that is displayed in the caption.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage CaptionImage { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionImage As IImage
```

Property Value

A **Microsoft.Maui.Graphics.IImage** instance.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.ConnectionStyle Property

Gets or sets a value indicating whether drawing a link from/to this node should connect an item or the whole node as an integral entity.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewConnectionStyle ConnectionStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property ConnectionStyle As TreeViewConnectionStyle
```

Property Value

A member of the TreeViewConnectionStyle enumeration. The default value is **Items**.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.EnableStyledText Property

Gets or sets a value indicating whether styled text rendering is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool EnableStyledText { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableStyledText As Boolean
```

Property Value

true to parse the text of the node's caption and items in order to find formatting tags; otherwise **false**. Initialized with the value of EnableStyledText.

Remarks

Styled text allows applying various attributes to the characters it contains. Attributes are specified using HTML-like formatting tags embedded in the raw text. Formatting tags are listed and described in the **Text Attributes and Formatting** section of this help file.

If **EnableStyledText** is set to **true**, the node's Caption and the items' Label are parsed and analyzed each time they change. Parsing determines what formatting tags are used and adds the corresponding attribute records to MvcDiagram internal representation of styled text. If the property value is **false**, the text is not processed in any way and is displayed without formatting.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.Font Property

Gets or sets the font used to render the node's caption.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Font Font { get; set; }
```

Visual Basic Copy Code

```
Public Overrides Property Font As Font
```

Property Value

A Font instance specifying the font attributes.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.HasItems Property

DiagramNode.HasItems override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override bool HasItems { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property HasItems As Boolean
```

Property Value

See Also

TreeViewNode Members
TreeViewNode Class

TreeViewNode.RootItems Property

Gets a list of the root TreeViewItem objects displayed in this node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IList<TreeViewItem> RootItems { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property RootItems As IList(Of TreeViewItem)
```

Property Value

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.ScrollPosition Property

Gets or sets the current scroll position.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ScrollPosition { get; set; }
```

Visual Basic Copy Code

```
Public Property ScrollPosition As Double
```

Property Value

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.Shape Property

Gets or sets the shape of the table's outline.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SimpleShape Shape { get; set; }
```

Visual Basic Copy Code

```
Public Property Shape As SimpleShape
```

Property Value

A member of the SimpleShape enumeration. The default is **Rectangle**.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.ShowLines Property

Gets or sets whether to display lines connecting child items to parent items.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ShowLines { get; set; }
```

Visual Basic Copy Code

```
Public Property ShowLines As Boolean
```

Property Value

true to draw connecting lines, or **false** otherwise. The default value is **true**.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.TextFormat Property

Gets or sets the alignment of texts displayed inside this shape node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringFormat TextFormat { get; set; }
```

Visual Basic Copy Code

```
Public Property TextFormat As StringFormat
```

Property Value

An instance of the StringFormat class.

Remarks

Use this property to specify the wrapping mode and alignment of the caption text.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNode.TextPadding Property

Gets or sets the spacing between the item boundaries and its text.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override Thickness TextPadding { get; set; }
```

Visual Basic Copy Code

```
Public Overrides Property TextPadding As Thickness
```

Property Value

A Thickness instance specifying the size of the padding space left around the text.

See Also

TreeViewNode Members
TreeViewNode Class
MindFusion.Diagramming Namespace

TreeViewNodeStyle Class

Represents a reusable set of properties that define the appearance of tree view nodes.

Syntax

C# Copy Code

```
public class TreeViewNodeStyle : DiagramNodeStyle
```

Visual Basic Copy Code

```
Public Class TreeViewNodeStyle
    Inherits DiagramNodeStyle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle
      MindFusion.Diagramming.DiagramNodeStyle
        MindFusion.Diagramming.TreeViewNodeStyle

See Also

TreeViewNodeStyle Members
MindFusion.Diagramming Namespace

TreeViewNodeStyle Members

Represents a reusable set of properties that define the appearance of tree view nodes.

The following tables list the members exposed by the TreeViewNodeStyle type.

Public Constructors

	Name	Description
	TreeViewNodeStyle	Initializes a new instance of the **TreeViewNodeStyle** class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of the item. (Inherited from DiagramItemStyle.)
	Effects	Gets a list of the effects applied to the node. (Inherited from DiagramNodeStyle.)
	FontFamily	Gets or sets the font family of the item text. (Inherited from DiagramItemStyle.)
	FontSize	Gets or sets the font size of the item text. (Inherited from DiagramItemStyle.)
	FontStyle	Gets or sets the font style of the item text. (Inherited from DiagramItemStyle.)
	FontUnit	Gets or sets the font unit of the item text. (Inherited from DiagramItemStyle.)
	ShadowBrush	Gets or sets the brush used to paint the shadow of the item. (Inherited from DiagramItemStyle.)
	Stroke	Gets or sets the brush used to paint the outline of the item. (Inherited from DiagramItemStyle.)
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items. (Inherited from DiagramItemStyle.)
	StrokeThickness	Gets or sets the thickness of the item's outline, expressed in the current units of measure. (Inherited from DiagramItemStyle.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItemStyle.)

Public Methods

	Name	Description
	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
S	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JsonObject. (Inherited from DiagramNodeStyle.)
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)

	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a JsonObject. (Inherited from DiagramNodeStyle.)
	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

TreeViewNodeStyle Class
MindFusion.Diagramming Namespace

TreeViewNodeStyle Constructor

Initializes a new instance of the TreeViewNodeStyle class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public TreeViewNodeStyle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

TreeViewNodeStyle Class
MindFusion.Diagramming Namespace

TreeViewNodeStyle Events

UndoEventArgs Class

Contains the arguments passed to the ActionRecorded, ActionUndone and ActionRedone event handlers.

Syntax

C# Copy Code

```
public class UndoEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class UndoEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.UndoEventArgs
      MindFusion.Diagramming.UndoValidationEventArgs

See Also

UndoEventArgs Members
MindFusion.Diagramming Namespace

UndoEventArgs Members

Contains the arguments passed to the ActionRecorded, ActionUndone and ActionRedone event handlers.

The following tables list the members exposed by the UndoEventArgs type.

Public Constructors

	Name	Description
	UndoEventArgs	Initializes a new instance of the **UndoEventArgs** class.

Public Properties

	Name	Description
	Command	Gets a reference to the Command instance related to the event being handled.

See Also

UndoEventArgs Class
MindFusion.Diagramming Namespace

UndoEventArgs Constructor

Initializes a new instance of the UndoEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public UndoEventArgs (
    Command command
)
```

Visual Basic Copy Code

```
Public New ( _
    command As Command _
)
```

Parameters

command
The Command related to the event.

See Also

UndoEventArgs Properties

UndoEventArgs.Command Property

Gets a reference to the **Command** instance related to the event being handled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Command Command { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Command As Command
```

Property Value

An instance of the **Command** class.

See Also

UndoEventArgs Members
UndoEventArgs Class
MindFusion.Diagramming Namespace

UndoManager Class

Implements undo/redo functionality in MvcDiagram.

Syntax

C# Copy Code

```
public class UndoManager
```

Visual Basic Copy Code

```
Public Class UndoManager
```

Remarks

The only instance of **UndoManager** can be accessed through the UndoManager property of Diagram.

Use the UndoEnabled property to enable or disable saving action records for later undo. You might need to disable undo before performing actions that must not be undoable by end-users, such as creating special start/end nodes.

MvcDiagram does not provide shortcuts to undo or redo an action. An application must provide the user interface means for undo/redo and call the Undo or Redo methods when necessary. Undoing or redoing an action updates the current record pointer of the undo-history queue. To get access to that queue, use the History property.

Sometimes several actions must be represented as a single atomic operation. That can be achieved by using composites, represented by the CompositeCmd class. However actions records that are implicitly created by MvcDiagram as a response to users actions or method calls cannot be added directly to a composite. Instead you must use the StartComposite method to create an 'active' composite, to which implicitly created records are added automatically.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.UndoManager

See Also

UndoManager Members
MindFusion.Diagramming Namespace

UndoManager Members

Implements undo/redo functionality in MvcDiagram.

The following tables list the members exposed by the UndoManager type.

Public Properties

	Name	Description
	History	Gets a reference to the undo manager's CommandHistory instance.
	UndoEnabled	Gets or sets a value indicating whether saving action records for later undo is enabled.

Public Methods

	Name	Description
	Redo	Executes again an action saved in the command history.
	StartComposite	Overloaded. Creates an 'active' composite record to store records of actions that are implicitly created by MvcDiagram.
	Undo	Undoes an action saved in the command history.

See Also

UndoManager Class
MindFusion.Diagramming Namespace

UndoManager Methods

UndoManager.Redo Method

Executes again an action saved in the command history.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Redo ()
```

Visual Basic Copy Code

```
Public Sub Redo ()
```

Remarks

Executes again a command from the command history. The action that would be redone by calling this method is pointed by NextRedo property of the CommandHistory class.

See Also

UndoManager Members
UndoManager Class

UndoManager.StartComposite Method

Creates an 'active' composite record to store records of actions that are implicitly created by MvcDiagram.

Overload List

Name	Description
UndoManager.StartComposite (String)	Creates an 'active' composite record with the specified title.
UndoManager.StartComposite (String, Boolean)	Creates an 'active' composite record with the specified settings.

See Also

UndoManager Members
UndoManager Class
MindFusion.Diagramming Namespace

UndoManager.StartComposite Method (String)

Creates an 'active' composite record with the specified title.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CompositeCmd StartComposite (
    string title
)
```

Visual Basic Copy Code

```
Public Function StartComposite( _
    title As String _
) As CompositeCmd
```

Parameters

title
A string to be associated with the composite record. It can be used later to give more information to users what exactly will be undone or redone. Use the Title property of the command class to get the title of an action record.

Return Value

An instance of the CompositeCmd class which represents the active composite record. Call its Execute method when done with recording actions in the composite.

Remarks

Sometimes several actions must be represented as a single atomic operation. That can be achieved by using composites, represented by the CompositeCmd class. However actions records that are implicitly created by MvcDiagram as a response to users actions or method calls cannot be added directly to a composite. The **StartComposite** method creates an 'active' composite, to which implicitly created records are added automatically. Call the composite's Execute method to stop adding action records to it and save it in the command history. Calling Undo or Redo for the composite record, undoes or redoes all actions contained within it.

See Also

StartComposite Method Overload List
UndoManager Members

UndoManager.StartComposite Method (String, Boolean)

Creates an 'active' composite record with the specified settings.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CompositeCmd StartComposite (
    string title,
    bool saveZOrder
)
```

Visual Basic Copy Code

```
Public Function StartComposite( _
    title As String, _
    saveZOrder As Boolean _
) As CompositeCmd
```

Parameters

title
A string to be associated with the composite record. It can be used later to give more information to users what exactly will be undone or redone. Use the Title property of the command class to get the title of an action record.

saveZOrder
Set this parameter to **true** to save the whole z-order array before executing the composite command, and to restore it back when the command is undone. Use that when there will be create or delete commands added to the composite one, because they do not guarantee restoring the items in the same z-order when the commands are undone or redone.

Return Value

An instance of the CompositeCmd class which represents the active composite record. Call its Execute method when done with recording actions in the composite.

Remarks

Sometimes several actions must be represented as a single atomic operation. That can be achieved by using composites, represented by the CompositeCmd class. However actions records that are implicitly created by MvcDiagram as a response to users actions or method calls cannot be added directly to a composite. The **StartComposite** method creates an 'active' composite, to which implicitly created records are added automatically. Call the composite's Execute method to stop adding action records to it and save it in the command history. Calling Undo or Redo for the composite record, undoes or redoes all actions contained within it.

Example

The following example illustrates how to create composite commands:

C# Copy Code

```
// Implicitly created undo records will be saved here
CompositeCmd composite = diagram.UndoManager.StartComposite("Create group");

// Call methods that create undo records
ShapeNode node1 = diagram.Factory.CreateShapeNode(0, 0, 30, 30);
ShapeNode node2 = diagram.Factory.CreateShapeNode(0, 0, 15, 15);
Group g = diagram.Factory.CreateGroup(node1);
g.AttachToCorner(node2, 0);

// This goes to the active composite too
ChangeItemCmd change = new ChangeItemCmd(node2, "");
node2.Locked = true;
change.Execute();

// Save the active composite in history
composite.Execute();
```

Visual Basic Copy Code

```
' Implicitly created undo records will be saved here
Dim composite As CompositeCmd = diagram.UndoManager.StartComposite("Create group")

' Call methods that create undo records
Dim node1 As ShapeNode = diagram.Factory.CreateShapeNode(0, 0, 30, 30)
Dim node2 As ShapeNode = diagram.Factory.CreateShapeNode(0, 0, 15, 15)
Dim g As Group = diagram.Factory.CreateGroup(node1)
g.AttachToCorner(node2, 0)

' This goes to the active composite too
Dim change As ChangeItemCmd = New ChangeItemCmd(node2, "")
node2.Locked = True
change.Execute()

' Save the active composite in history
composite.Execute()
```

See Also

StartComposite Method Overload List
UndoManager Members
UndoManager Class
MindFusion.Diagramming Namespace

UndoManager.Undo Method

Undoes an action saved in the command history.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Undo ()
```

Visual Basic Copy Code

```
Public Sub Undo ()
```

Remarks

Undoes a command from the command history. The action that would be undone by calling this method is pointed by NextUndo property of the CommandHistory class.

See Also

UndoManager Members
UndoManager Class

UndoManager Properties

UndoManager.History Property

Gets a reference to the undo manager's CommandHistory instance.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public CommandHistory History { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property History As CommandHistory
```

Property Value

An instance of the CommandHistory class.

Remarks

History represents the undo manager's queue of action records. This queue provides access to the NextUndo and NextRedo records. You could set the queue's Capacity, specifying how many records can be added to the queue before the oldest ones start to drop off its end.

See Also

UndoManager Members
UndoManager Class
MindFusion.Diagramming Namespace
Command class

UndoManager.UndoEnabled Property

Gets or sets a value indicating whether saving action records for later undo is enabled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool UndoEnabled { get; set; }
```

Visual Basic Copy Code

```
Public Property UndoEnabled As Boolean
```

Property Value

true to save actions for later undo; otherwise **false**. The default is **false**.

Remarks

You might need to disable undo:

- before executing actions you don't want end-users to undo, for example creating special start/end nodes.
- before creating programmatically a lot of new diagram items - say hundreds or thousands. MvcDiagram creates an undo record for each item creation; if the Capacity of the command history is smaller than the number of items created, most of

the records will drop from the history queue while adding new ones. That would waste processing power and memory for creating undo records that won't be used.

See Also

UndoManager Members
UndoManager Class
MindFusion.Diagramming Namespace

UndoValidationEventArgs Class

Contains the arguments passed to the ActionRecording event handlers.

Syntax

C# Copy Code

```
public class UndoValidationEventArgs : UndoEventArgs
```

Visual Basic Copy Code

```
Public Class UndoValidationEventArgs
    Inherits UndoEventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.UndoEventArgs
      MindFusion.Diagramming.UndoValidationEventArgs

See Also

UndoValidationEventArgs Members
MindFusion.Diagramming Namespace

UndoValidationEventArgs Members

Contains the arguments passed to the ActionRecording event handlers.

The following tables list the members exposed by the UndoValidationEventArgs type.

Public Constructors

	Name	Description
	UndoValidationEventArgs	Initializes a new instance of the **UndoValidationEventArgs** class.

Public Properties

	Name	Description
	Cancel	Gets or sets a value value indicating whether the operation should be canceled.
	Command	Gets a reference to the Command instance related to the event being handled. (Inherited from UndoEventArgs.)

See Also

UndoValidationEventArgs Class
MindFusion.Diagramming Namespace

UndoValidationEventArgs Constructor

Initializes a new instance of the UndoValidationEventArgs class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public UndoValidationEventArgs (
    Command command
)
```

Visual Basic Copy Code

```
Public New ( _
    command As Command _
)
```

Parameters

command
The Command related to the event.

See Also

UndoValidationEventArgs Class
MindFusion.Diagramming Namespace

UndoValidationEventArgs Properties

UndoValidationEventArgs.Cancel Property

Gets or sets a value value indicating whether the operation should be canceled.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool Cancel { get; set; }
```

Visual Basic Copy Code

```
Public Property Cancel As Boolean
```

Property Value

true to cancel the operation; **false** to approve it.

See Also

UndoValidationEventArgs Members
UndoValidationEventArgs Class
MindFusion.Diagramming Namespace

UnitType Enumeration

Defines values that indicate whether point positions are specified as percentage, or in absolute units relative to the node's top-left corner.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum UnitType
```

Visual Basic Copy Code

```
Public Enum UnitType
```

Members

	Member name	Description
	Percentage	Point coordinates are specified as percentage of the node's size.
	Fixed	Point coordinates designate fixed offset from the node's top-left corner.

See Also

MindFusion.Diagramming Namespace
XUnit Property
YUnit Property

Vector Structure

Represents a two-dimensional vector.

Syntax

C# Copy Code

```
public struct Vector
```

Visual Basic Copy Code

```
Public Structure Vector
```

See Also

Vector Members
MindFusion.Diagramming Namespace

Vector Members

Represents a two-dimensional vector.

The following tables list the members exposed by the Vector type.

Public Constructors

	Name	Description
	Vector	Overloaded.

Public Properties

	Name	Description
	Length	Returns the vector length.

	LengthSquared	Returns the squared length.
	X	X coordinate of the vector.
	Y	Y coordinate of the vector.

Public Methods

	Name	Description
	Add	Overloaded.
	AngleBetween	Returns the angle between two vectors.
	Divide	Divides a vector by a scalar value.
	Equals	Overloaded.
	GetHashCode	Object.GetHashCode override.
	Left	Returns the left vector.
	Multiply	Overloaded.
	Negate	Reverses this vector's direction.
	Normalize	Normalizes this vector.
	Right	Returns the right vector.
	Subtract	Subtracts two vectors.
	ToString	Overloaded.

See Also

Vector Structure
MindFusion.Diagramming Namespace

Vector Constructor

Overload List

Name	Description
Vector (Double, Double)	Initializes a new instance of the Vector structure with specified coordinates.
Vector (Point, Point)	Initializes a new instance of the Vector structure from the specified points.

See Also

Vector Structure
MindFusion.Diagramming Namespace

Vector Constructor (Double, Double)

Initializes a new instance of the Vector structure with specified coordinates.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#	Copy Code

```
public Vector (
    double x,
    double y
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Double, _
    y As Double _
)
```

Parameters

x
X coordinate of the vector.

y
Y coordinate of the vector.

See Also

Vector Constructor Overload List
Vector Structure
MindFusion.Diagramming Namespace

Vector Constructor (Point, Point)

Initializes a new instance of the Vector structure from the specified points.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Vector (
    Point p1,
    Point p2
)
```

Visual Basic Copy Code

```
Public New ( _
    p1 As Point, _
    p2 As Point _
)
```

Parameters

p1
Origin point of the vector.

p2
End point of the vector.

See Also

Vector Constructor Overload List
Vector Structure
MindFusion.Diagramming Namespace

Vector Methods

Vector.Add Method

Overload List

Name	Description
Vector.Add (Vector, Point)	Adds vector to a point.
Vector.Add (Vector, Vector)	Adds two vectors.

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Add Method (Vector, Point)

Adds vector to a point.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Point Add (
    Vector vector,
    Point point
)
```

Visual Basic Copy Code

```
Public Shared Function Add( _
    vector As Vector, _
    point As Point _
) As Point
```

Parameters

vector
A Vector instance.

point
A **Microsoft.Maui.Graphics.Point** instance.

Return Value

A point offset from the specified point by adding specified vector.

See Also

Add Method Overload List
Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Add Method (Vector, Vector)

Adds two vectors.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Vector Add (
    Vector vector1,
    Vector vector2
)
```

Visual Basic Copy Code

```
Public Shared Function Add( _
    vector1 As Vector, _
    vector2 As Vector _
) As Vector
```

Parameters

vector1

vector2

Return Value

See Also

Add Method Overload List
Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.AngleBetween Method

Returns the angle between two vectors.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double AngleBetween (
    Vector vector1,
    Vector vector2
)
```

Visual Basic Copy Code

```
Public Shared Function AngleBetween( _
    vector1 As Vector, _
    vector2 As Vector _
) As Double
```

Parameters

vector1
A Vector instance.

vector2
A Vector instance.

Return Value

The angle between specified vectors.

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Divide Method

Divides a vector by a scalar value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Vector Divide (
    Vector vector,
    double scalar
)
```

Visual Basic Copy Code

```
Public Shared Function Divide( _
    vector As Vector, _
    scalar As Double _
) As Vector
```

Parameters

vector
A Vector instance.

scalar
A **double** value.

Return Value

A Vector with same direction and divided length.

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Equals Method

Overload List

Name	Description
Vector.Equals (Object)	Object.Equals override.
Vector.Equals (Vector)	Compare this vector with the specified vector for equality.
Vector.Equals (Vector, Vector)	Compares two vectors for equality.

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Equals Method (Object)

Object.Equals override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override bool Equals (
    Object o
)
```

Visual Basic Copy Code

```
Public Overrides Function Equals( _
    o As Object _
) As Boolean
```

Parameters

o
The object to compare this vector with.

Return Value

true if the objects are equal, or otherwise false.

See Also

Equals Method Overload List
Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Equals Method (Vector)

Compare this vector with the specified vector for equality.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public new bool Equals (
    Vector value
)
```

Visual Basic Copy Code

```
Public Shadows Function Equals( _
    value As Vector _
) As Boolean
```

Parameters

value
The vector to compare with.

Return Value

true if the vectors are equal, or otherwise false.

See Also

Equals Method Overload List
Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Equals Method (Vector, Vector)

Compares two vectors for equality.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public new static bool Equals (
    Vector vector1,
    Vector vector2
)
```

Visual Basic Copy Code

```
Public Shadows Shared Function Equals( _
    vector1 As Vector, _
    vector2 As Vector _
) As Boolean
```

Parameters

vector1
The first vector to compare.

vector2
The second vector to compare.

Return Value

true if the vectors are equal, or otherwise false.

See Also

Equals Method Overload List
Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.GetHashCode Method

Object.GetHashCode override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public override int GetHashCode ()
```

Visual Basic Copy Code

```
Public Overrides Function GetHashCode () As Integer
```

Return Value

The vector's hash code.

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Left Method

Returns the left vector.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Vector Left (
    Vector v
)
```

Visual Basic Copy Code

```
Public Shared Function Left( _
    v As Vector _
) As Vector
```

Parameters

v

Return Value

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Multiply Method

Overload List

Name	Description
Vector.Multiply (Double, Vector)	Multiplies a vector by a scalar value.
Vector.Multiply (Vector, Double)	Multiplies a vector by a scalar value.
Vector.Multiply (Vector, Vector)	Multiplies two vectors.

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Multiply Method (Double, Vector)

Multiplies a vector by a scalar value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Vector Multiply (
    double scalar,
```

```
    Vector vector
)
```

Visual Basic Copy Code

```
Public Shared Function Multiply( _
    scalar As Double, _
    vector As Vector _
) As Vector
```

Parameters

scalar

A **double** value.

vector

A Vector instance.

Return Value

A Vector with same direction and multiplied length.

See Also

Multiply Method Overload List
Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Multiply Method (Vector, Double)

Multiplies a vector by a scalar value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Vector Multiply (
    Vector vector,
    double scalar
)
```

Visual Basic Copy Code

```
Public Shared Function Multiply( _
    vector As Vector, _
    scalar As Double _
) As Vector
```

Parameters

vector

A Vector instance.

scalar

A **double** value.

Return Value

A Vector with same direction and multiplied length.

See Also

Multiply Method Overload List
Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Multiply Method (Vector, Vector)

Multiplies two vectors.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static double Multiply (
    Vector vector1,
    Vector vector2
)
```

Visual Basic Copy Code

```
Public Shared Function Multiply( _
    vector1 As Vector, _
    vector2 As Vector _
) As Double
```

Parameters

vector1
A Vector instance.

vector2
A Vector instance.

Return Value

The dot product of specified vectors.

See Also

Multiply Method Overload List
Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Negate Method

Reverses this vector's direction.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Negate ()
```

Visual Basic Copy Code

```
Public Sub Negate ()
```

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Normalize Method

Normalizes this vector.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public void Normalize ()
```

Visual Basic Copy Code

```
Public Sub Normalize ()
```

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Right Method

Returns the right vector.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static Vector Right (
    Vector v
)
```

Visual Basic Copy Code

```
Public Shared Function Right( _
    v As Vector _
) As Vector
```

Parameters

v

Return Value

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Subtract Method

Subtracts two vectors.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public static Vector Subtract (
    Vector vector1,
    Vector vector2
)
```

Visual Basic

```
Public Shared Function Subtract( _
    vector1 As Vector, _
    vector2 As Vector _
) As Vector
```

Parameters

vector1
A Vector instance.

vector2
A Vector instance.

Return Value

The difference between specified vectors.

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.ToString Method

Overload List

Name	Description
Vector.ToString ()	Object.ToString override.
Vector.ToString (IFormatProvider)	Returns the vector's string representation in specified culture.

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.ToString Method ()

Object.ToString override.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public override string ToString ()
```

Visual Basic

```
Public Overrides Function ToString () As String
```

Return Value

The vector's string representation.

See Also

ToString Method Overload List
Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.ToString Method (IFormatProvider)

Returns the vector's string representation in specified culture.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ToString (
    IFormatProvider provider
)
```

Visual Basic Copy Code

```
Public Function ToString( _
    provider As IFormatProvider _
) As String
```

Parameters

provider
An IFormatProvider (e.g. CultureInfo) specifying number format.

Return Value

The vector's string representation.

See Also

ToString Method Overload List
Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector Properties

Vector.Length Property

Returns the vector length.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Length { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Length As Double
```

Property Value

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.LengthSquared Property

Returns the squared length.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double LengthSquared { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property LengthSquared As Double
```

Property Value

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.X Property

X coordinate of the vector.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double X { get; set; }
```

Visual Basic Copy Code

```
Public Property X As Double
```

Property Value

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

Vector.Y Property

Y coordinate of the vector.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double Y { get; set; }
```

Visual Basic Copy Code

```
Public Property Y As Double
```

Property Value

See Also

Vector Members
Vector Structure
MindFusion.Diagramming Namespace

VideoNode Class

Represents nodes that render video stream assigned to the VideoLocation property.

Syntax

C# Copy Code

```
public class VideoNode : DiagramNode
```

Visual Basic Copy Code

```
Public Class VideoNode
    Inherits DiagramNode
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.VideoNode

See Also

VideoNode Members
MindFusion.Diagramming Namespace

VideoNode Members

Represents nodes that render video stream assigned to the VideoLocation property.

The following tables list the members exposed by the VideoNode type.

Public Constructors

	Name	Description
	VideoNode	Overloaded.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)

	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)
	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)

	VideoLocation	Gets or sets the URL of the video stream displayed in this VideoNode.
	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	Clone	Creates a copy of this item. (Inherited from DiagramItem.)
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)

	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)
	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)

	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)
	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)

	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)
	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFrom	Deserializes the object from the specified reader.
	LoadFromJson	Loads the item content from a JsonObject.
	LoadFromXml	Loads the item content from an XML element.
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)

	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)
	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveTo	Serializes the object to the specified writer.
	SaveToJson	Saves the item content into a JsonObject.
	SaveToXml	Saves the item content into an XML element.
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

VideoNode Class
MindFusion.Diagramming Namespace

VideoNode Constructor

Overload List

Name	Description
VideoNode ()	Initializes a new instance of the VideoNode class.
VideoNode (Diagram)	Initializes a new instance of the VideoNode class with default values supplied form the specified Diagram.
VideoNode (VideoNode)	Initializes a new instance of the VideoNode class using the specified node as a prototype.

See Also

VideoNode Class
MindFusion.Diagramming Namespace

VideoNode Constructor ()

Initializes a new instance of the VideoNode class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public VideoNode ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

VideoNode Constructor Overload List
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode Constructor (Diagram)

Initializes a new instance of the VideoNode class with default values supplied form the specified Diagram.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public VideoNode (
    Diagram parent
)
```

Visual Basic Copy Code

```
Public New ( _
    parent As Diagram _
)
```

Parameters

parent
The Diagram from which to obtain default values for the node properties.

See Also

VideoNode Constructor Overload List
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode Constructor (VideoNode)

Initializes a new instance of the VideoNode class using the specified node as a prototype.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public VideoNode (
    VideoNode prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As VideoNode _
)
```

Parameters

prototype
The node to use as a prototype. The properties of the new node are initialized from the property values of this node.

See Also

VideoNode Constructor Overload List
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode Methods

VideoNode.LoadFrom Method

Deserializes the object from the specified reader.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFrom (
    BinaryReader reader,
    PersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFrom( _
    reader As BinaryReader, _
    context As PersistContext _
)
```

Parameters

reader
A BinaryReader object from which to read the item's data.

context
A PersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode.LoadFromJson Method

Loads the item content from a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the item's data.

context
A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode.LoadFromXml Method

Loads the item content from an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
```

```
    context As XmlPersistContext _
)
```

Parameters

xmlElement
An XmlElement containing the item's data.

context
An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode.SaveTo Method

Serializes the object to the specified writer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveTo (
    BinaryWriter writer,
    PersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveTo( _
    writer As BinaryWriter, _
    context As PersistContext _
)
```

Parameters

writer
A BinaryWriter object the item's data should be written to.

context
A PersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode.SaveToJson Method

Saves the item content into a JsonObject.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
```

```
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

A JsonObject the item's data should be stored into.

context

A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode.SaveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

An XmlElement the item's data should be stored into.

context

An XmlPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

VideoNode Properties

VideoNode.VideoLocation Property

Gets or sets the URL of the video stream displayed in this VideoNode.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string VideoLocation { get; set; }
```

Visual Basic Copy Code

```
Public Property VideoLocation As String
```

Property Value

See Also

VideoNode Members
VideoNode Class
MindFusion.Diagramming Namespace

View3D Enumeration

Defines keys used to associate 3D position data with a node through the LayoutTraits property.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public enum View3D
```

Visual Basic Copy Code

```
Public Enum View3D
```

Members

	Member name	Description
	Position	Specifies the position of a node in 3D coordinate space. The value associated with this key should be a Point3D object.

See Also

MindFusion.Diagramming Namespace

VisitNode Delegate

Defines the prototype for methods called by VisitNodes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public delegate bool VisitNode (
    DiagramNode node,
    DiagramItem relatedItem
)
```

Visual Basic Copy Code

```
Public Delegate Function VisitNode( _
    node As DiagramNode, _
    relatedItem As DiagramItem _
) As Boolean
```

Parameters

node

relatedItem

Return Value

See Also

MindFusion.Diagramming Namespace

XmlOptions Class

Defines properties used to customize the XML serialization process.

Syntax

C# Copy Code

```
public class XmlOptions
```

Visual Basic Copy Code

```
Public Class XmlOptions
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.XmlOptions

See Also

XmlOptions Members
MindFusion.Diagramming Namespace

XmlOptions Members

Defines properties used to customize the XML serialization process.

The following tables list the members exposed by the XmlOptions type.

Public Constructors

	Name	Description
	XmlOptions	Initializes a new instance of the XmlOptions class.

Public Properties

	Name	Description

	UseXmlSerializer	Gets or sets whether to use XmlSerializer for non-primitive object properties.

See Also

XmlOptions Class
MindFusion.Diagramming Namespace

XmlOptions Constructor

Initializes a new instance of the XmlOptions class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlOptions ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

XmlOptions Class
MindFusion.Diagramming Namespace

XmlOptions Properties

XmlOptions.UseXmlSerializer Property

Gets or sets whether to use XmlSerializer for non-primitive object properties.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool UseXmlSerializer { get; set; }
```

Visual Basic Copy Code

```
Public Property UseXmlSerializer As Boolean
```

Property Value

See Also

XmlOptions Members
XmlOptions Class
MindFusion.Diagramming Namespace

XmlPersistContext Class

Contains methods that can help you serialize and deserialize MvcDiagram items to end from XML documents.

Syntax

C# Copy Code

```
public class XmlPersistContext
```

Visual Basic Copy Code

```
Public Class XmlPersistContext
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.XmlPersistContext

See Also

XmlPersistContext Members
MindFusion.Diagramming Namespace

XmlPersistContext Members

Contains methods that can help you serialize and deserialize MvcDiagram items to end from XML documents.

The following tables list the members exposed by the XmlPersistContext type.

Public Constructors

	Name	Description
	XmlPersistContext	Initializes a new instance of the **XmlPersistContext** class.

Public Fields

	Name	Description
	LatestFormat	Specifies the latest file format revision number.

Public Properties

	Name	Description
	BaseURI	Gets or sets the base URI of the XML document being loaded or saved. When not specified, the BaseURI of the underlying XmlDocument is used.
	ClassVersion	Gets the class version of the currently deserialized item.
	Diagram	Gets the underlying diagram object.
	FileVersion	Gets the format revision number for the file being currently serialized.
	IncludeUnalteredProperties	Gets or sets a value specifies whether properties that have their default value should be serialized to XML.
	XmlDocument	Gets the underlying XML document.
	XmlOptions	An XmlOptions object providing properties for customizing the XML serialization process.

Public Methods

	Name	Description
	AddChildElement	Overloaded.

	FromRelativePath	Converts the specified path from relative to the BaseURI to an absolute path.
	ReadAnchorPattern	Reads an AnchorPattern with the specified name.
	ReadBool	Overloaded.
	ReadBrush	Overloaded.
	ReadByte	Reads a byte value with the specified name.
	ReadChar	Reads a char value with the specified name.
	ReadColor	Overloaded.
	ReadColorArray	Reads a color array with the specified name.
	ReadDecimal	Reads a decimal value with the specified name.
	ReadDouble	Reads a double value with the specified name.
	ReadEffect	Overloaded. Reads a NodeEffect object.
	ReadEnum	Overloaded.
	ReadFloat	Overloaded.
	ReadFloatArray	Reads a float array with the specified name.
	ReadFont	Overloaded.
	ReadGroup	Reads a group object with the specified name.
	ReadHashtable	Reads a hash or traits table object with the specified name.
	ReadImage	Reads an Image object with the specified name.
	ReadInt	Overloaded.
	ReadItemReference	Overloaded.
	ReadLong	Reads a long value with the specified name.
	ReadMap	Reads a **MapContent** object with the specified name.
	ReadNodeConstraints	Reads a NodeConstraints object with the specified name.
	ReadObject	Reads an object with the specified name.
	ReadPen	Overloaded. Reads a MindFusion.Drawing.Pen object with the specified name.
	ReadPointCollection	Reads a PointCollection object with the specified name.
	ReadPointCoordinates	Reads a List<Point> object with the specified name.
	ReadPointF	Overloaded.
	ReadRectangleF	Reads a Rect object with the specified name.
	ReadSByte	Reads a 8-bit signed integer value with the specified name.
	ReadShape	Reads a Shape object with the specified name.
	ReadShort	Overloaded.
	ReadSizeF	Reads a Size object with the specified name.
	ReadStream	Reads a stream with the specified name.
	ReadStreamOrXml	Reads a stream or xml.
	ReadString	Overloaded.

	ReadStringFormat	Overloaded.
	ReadStyle	Reads a Style object with the specified name.
	ReadSvg	Reads a SVG object with the specified name.
	ReadTag	Reads the specified tag for the specified object.
	ReadTheme	Reads a Theme object with the specified name.
	ReadThickness	Reads a Thickness object with the specified name.
	ReadUInt	Reads an unsigned integer value with the specified name.
	ReadULong	Reads an unsigned long value with the specified name.
	ReadUShort	Reads an unsigned short value with the specified name.
	ReadVisioContent	Reads a Visio content with the specified name.
	ReadXmlObject	Reads an object using XmlSerializer.
	SelectNode	Returns the child node with the specified name.
	SelectNodes	Returns the child nodes with the specified name.
	ToRelativePath	Converts the specified absolute path to a path relative to the BaseURI.
	WriteAnchorPattern	Writes the specified anchor pattern.
	WriteBool	Writes a boolean value with the specified name.
	WriteBrush	Writes a reference to the specified brush. The brush is registered within the internal hashtable for subsequent serialization.
	WriteByte	Write a byte value with the specified name.
	WriteChar	Write a char value with the specified name.
	WriteColor	Writes the specified color.
	WriteColorArray	Writes the specified array.
	WriteDecimal	Write a decimal value with the specified name.
	WriteDouble	Writes a double value with the specified name.
	WriteEffect	Writes a reference to the specified effect. The effect is registered within the internal hashtable for subsequent serialization.
	WriteEnum	Writes the specified enum value.
	WriteFloat	Writes a float value with the specified name.
	WriteFloatArray	Writes the specified array.
	WriteFont	Writes the specified font.
	WriteGroup	Writes the specified group object.
	WriteHashtable	Writes the specified hashtable.
	WriteImage	Writes a reference to the specified image. The image is registered within the internal hashtable for subsequent serialization.
	WriteInt	Writes an integer value with the specified name.
	WriteItemReference	Writes a reference to the specified item with the specified name.
	WriteLong	Writes a long value with the specified name.
	WriteMap	Writes a reference to the specified **MapContent**. The map is registered within the internal hashtable for subsequent serialization.

	WriteObject	Writes the specified object under the specified name.
	WritePen	Writes the specified pen.
	WritePointCollection	Writes the specified point collection.
	WritePointCoordinates	Writes the specified List<Point>.
	WritePointF	Overloaded.
	WriteRectangleF	Writes a Rect with the specified name.
	WriteSByte	Write a 8-bit signed integer value with the specified name.
	WriteShape	Writes the specified shape with the specified name.
	WriteShort	Writes a short value with the specified name.
	WriteSizeF	Writes a Size with the specified name.
	WriteStream	Writes the specified stream.
	WriteString	Writes a string value with the specified name.
	WriteStringFormat	Writes the specified StringFormat object.
	WriteStyle	Writes a reference to the specified style. The style is registered within the internal hashtable for subsequent serialization.
	WriteSvg	Writes a reference to the specified SVG. The SVG is registered within the internal hashtable for subsequent serialization.
	WriteTag	Writes the specified tag of the specified object.
	WriteTheme	Writes the specified **Theme** object.
	WriteThickness	Writes a Thickness with the specified name.
	WriteTraits	Writes the specified traits hashtable.
	WriteUInt	Writes an unsigned integer value with the specified name.
	WriteULong	Writes an unsigned long value with the specified name.
	WriteUShort	Writes an unsigned short value with the specified name.
	WriteVisioContent	Writes a reference to the specified Visio content. The content is registered within the internal hashtable for subsequent serialization.
	WriteXmlObject	

See Also

XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext Fields

XmlPersistContext.LatestFormat Field

Specifies the latest file format revision number.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public static int LatestFormat
```

Visual Basic Copy Code

```
Public Shared LatestFormat As Integer
```

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext Constructor

Initializes a new instance of the XmlPersistContext class.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlPersistContext (
    Diagram diagram,
    XmlDocument document,
    int fileVersion
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram, _
    document As XmlDocument, _
    fileVersion As Integer _
)
```

Parameters

diagram
The Diagram that will be serialized or deserialized.

document
An **XmlDocument** instance representing the XML file.

fileVersion
Specifies the current file format number.

See Also

XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext Methods

XmlPersistContext.AddChildElement Method

Overload List

Name	Description
XmlPersistContext.AddChildElement (String, XmlElement)	Adds a new child node with the specified name to the specified parent node.

XmlPersistContext.AddChildElement (String, XmlElement, String)	Adds a new child node with the specified name and value to the specified parent node.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.AddChildElement Method (String, XmlElement)

Adds a new child node with the specified name to the specified parent node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement AddChildElement (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function AddChildElement( _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

elementName

parentElement

Return Value

See Also

AddChildElement Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.AddChildElement Method (String, XmlElement, String)

Adds a new child node with the specified name and value to the specified parent node.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement AddChildElement (
    string elementName,
    XmlElement parentElement,
    string innerText
)
```

Visual Basic Copy Code

```
Public Function AddChildElement( _
    elementName As String, _
    parentElement As XmlElement, _
    innerText As String _
) As XmlElement
```

Parameters

elementName

parentElement

innerText

Return Value

See Also

AddChildElement Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.FromRelativePath Method

Converts the specified path from relative to the BaseURI to an absolute path.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string FromRelativePath (
    string path
)
```

Visual Basic Copy Code

```
Public Function FromRelativePath( _
    path As String _
) As String
```

Parameters

path

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadAnchorPattern Method

Reads an AnchorPattern with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public AnchorPattern ReadAnchorPattern (
    string elementName,
```

```
    XmlElement parentElement,
    bool standalone
)
```

Visual Basic Copy Code

```
Public Function ReadAnchorPattern( _
    elementName As String, _
    parentElement As XmlElement, _
    standalone As Boolean _
) As AnchorPattern
```

Parameters

elementName

parentElement

standalone

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadBool Method

Overload List

Name	Description
XmlPersistContext.ReadBool (String, XmlElement)	Reads a boolean value with the specified name.
XmlPersistContext.ReadBool (String, XmlElement, Boolean)	Reads a boolean value with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadBool Method (String, XmlElement)

Reads a boolean value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ReadBool (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadBool( _
    elementName As String, _
```

```
    parentElement As XmlElement _
) As Boolean
```

Parameters

elementName

parentElement

Return Value

See Also

ReadBool Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadBool Method (String, XmlElement, Boolean)

Reads a boolean value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool ReadBool (
    string elementName,
    XmlElement parentElement,
    bool defaultValue
)
```

Visual Basic Copy Code

```
Public Function ReadBool( _
    elementName As String, _
    parentElement As XmlElement, _
    defaultValue As Boolean _
) As Boolean
```

Parameters

elementName

parentElement

defaultValue

Return Value

See Also

ReadBool Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadBrush Method

Overload List

Name	Description
XmlPersistContext.ReadBrush (String, XmlElement)	Reads a MindFusion.Drawing.Brush object with the specified name.

XmlPersistContext.ReadBrush (String, XmlElement, Brush)	Reads a MindFusion.Drawing.Brush object with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadBrush Method (String, XmlElement)

Reads a MindFusion.Drawing.Brush object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush ReadBrush (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadBrush( _
    elementName As String, _
    parentElement As XmlElement _
) As Brush
```

Parameters

elementName

parentElement

Return Value

See Also

ReadBrush Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadBrush Method (String, XmlElement, Brush)

Reads a MindFusion.Drawing.Brush object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Brush ReadBrush (
    string elementName,
    XmlElement parentElement,
    Brush defaultValue
)
```

Visual Basic Copy Code

```
Public Function ReadBrush( _
    elementName As String, _
    parentElement As XmlElement, _
    defaultValue As Brush _
) As Brush
```

Parameters

elementName

parentElement

defaultValue

Return Value

See Also

ReadBrush Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadByte Method

Reads a byte value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public byte ReadByte (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadByte( _
    elementName As String, _
    parentElement As XmlElement _
) As Byte
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadChar Method

Reads a char value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public char ReadChar (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic

```
Public Function ReadChar( _
    elementName As String, _
    parentElement As XmlElement _
) As Char
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadColor Method

Overload List

Name	Description
XmlPersistContext.ReadColor (String, XmlElement)	Reads a color with the specified name.
XmlPersistContext.ReadColor (String, XmlElement, Color)	Reads a color with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadColor Method (String, XmlElement)

Reads a color with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public Color ReadColor (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic

```
Public Function ReadColor( _
    elementName As String, _
```

```
    parentElement As XmlElement _
) As Color
```

Parameters

elementName

parentElement

Return Value

See Also

ReadColor Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadColor Method (String, XmlElement, Color)

Reads a color with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Color ReadColor (
    string elementName,
    XmlElement parentElement,
    Color defaultValue
)
```

Visual Basic Copy Code

```
Public Function ReadColor( _
    elementName As String, _
    parentElement As XmlElement, _
    defaultValue As Color _
) As Color
```

Parameters

elementName

parentElement

defaultValue

Return Value

See Also

ReadColor Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadColorArray Method

Reads a color array with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

Copy Code

```
public Color[] ReadColorArray (
    string elementName,
    string subElementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadColorArray( _
    elementName As String, _
    subElementName As String, _
    parentElement As XmlElement _
) As Color()
```

Parameters

elementName

subElementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadDecimal Method

Reads a decimal value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public decimal ReadDecimal (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadDecimal( _
    elementName As String, _
    parentElement As XmlElement _
) As Decimal
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadDouble Method

Reads a double value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ReadDouble (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadDouble( _
    elementName As String, _
    parentElement As XmlElement _
) As Double
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadEffect Method

Reads a NodeEffect object.

Overload List

Name	Description
XmlPersistContext.ReadEffect (String, XmlElement)	Reads a **NodeEffect** object with the specified name.
XmlPersistContext.ReadEffect (XmlElement)	Reads a **NodeEffect** object from the specified element.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadEffect Method (String, XmlElement)

Reads a NodeEffect object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeEffect ReadEffect (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadEffect( _
    elementName As String, _
    parentElement As XmlElement _
) As NodeEffect
```

Parameters

elementName

parentElement

Return Value

See Also

ReadEffect Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadEffect Method (XmlElement)

Reads a NodeEffect object from the specified element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeEffect ReadEffect (
    XmlElement element
)
```

Visual Basic Copy Code

```
Public Function ReadEffect( _
    element As XmlElement _
) As NodeEffect
```

Parameters

element

Return Value

See Also

ReadEffect Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadEnum Method

Overload List

Name	Description
XmlPersistContext.ReadEnum (String, XmlElement)	Reads an enum value with the specified name.
XmlPersistContext.ReadEnum (String, XmlElement, Object)	Reads an enum value with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadEnum Method (String, XmlElement)

Reads an enum value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object ReadEnum (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadEnum( _
    elementName As String, _
    parentElement As XmlElement _
) As Object
```

Parameters

elementName

parentElement

Return Value

See Also

ReadEnum Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadEnum Method (String, XmlElement, Object)

Reads an enum value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object ReadEnum (
    string elementName,
    XmlElement parentElement,
    Object defaultValue
)
```

Visual Basic Copy Code

```
Public Function ReadEnum( _
    elementName As String, _
    parentElement As XmlElement, _
    defaultValue As Object _
) As Object
```

Parameters

elementName

parentElement

defaultValue

Return Value

See Also

ReadEnum Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadFloat Method

Overload List

Name	Description
XmlPersistContext.ReadFloat (String, XmlElement)	Reads a float value with the specified name.
XmlPersistContext.ReadFloat (String, XmlElement, Double)	Reads a float value with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadFloat Method (String, XmlElement)

Reads a float value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double ReadFloat (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic

```
Public Function ReadFloat( _
    elementName As String, _
    parentElement As XmlElement _
) As Double
```

Parameters

elementName

parentElement

Return Value

See Also

ReadFloat Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadFloat Method (String, XmlElement, Double)

Reads a float value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public double ReadFloat (
    string elementName,
    XmlElement parentElement,
    double defaultValue
)
```

Visual Basic

```
Public Function ReadFloat( _
    elementName As String, _
    parentElement As XmlElement, _
    defaultValue As Double _
) As Double
```

Parameters

elementName

parentElement

defaultValue

Return Value

See Also

ReadFloat Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadFloatArray Method

Reads a float array with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public double[] ReadFloatArray (
    string elementName,
    string subElementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadFloatArray( _
    elementName As String, _
    subElementName As String, _
    parentElement As XmlElement _
) As Double()
```

Parameters

elementName

subElementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadFont Method

Overload List

Name	Description
XmlPersistContext.ReadFont (String, XmlElement)	Reads a Font object with the specified name.
XmlPersistContext.ReadFont (String, XmlElement, Font)	Reads a Font object with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadFont Method (String, XmlElement)

Reads a Font object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Font ReadFont (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadFont( _
    elementName As String, _
    parentElement As XmlElement _
) As Font
```

Parameters

elementName

parentElement

Return Value

See Also

ReadFont Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadFont Method (String, XmlElement, Font)

Reads a Font object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Font ReadFont (
    string elementName,
    XmlElement parentElement,
    Font defaultFont
)
```

Visual Basic Copy Code

```
Public Function ReadFont( _
    elementName As String, _
    parentElement As XmlElement, _
    defaultFont As Font _
) As Font
```

Parameters

elementName

parentElement

defaultFont

Return Value

See Also

ReadFont Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadGroup Method

Reads a group object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Group ReadGroup (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadGroup( _
    elementName As String, _
    parentElement As XmlElement _
) As Group
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadHashtable Method

Reads a hash or traits table object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Hashtable ReadHashtable (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadHashtable( _
    elementName As String, _
    parentElement As XmlElement _
) As Hashtable
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext.ReadImage Method

Reads an Image object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public IImage ReadImage (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadImage( _
    elementName As String, _
    parentElement As XmlElement _
) As IImage
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadInt Method

Overload List

Name	Description
XmlPersistContext.ReadInt (String, XmlElement)	Reads an integer value with the specified name.
XmlPersistContext.ReadInt (String, XmlElement, Int32)	Reads an integer value with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadInt Method (String, XmlElement)

Reads an integer value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int ReadInt (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadInt( _
    elementName As String, _
    parentElement As XmlElement _
) As Integer
```

Parameters

elementName

parentElement

Return Value

See Also

ReadInt Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadInt Method (String, XmlElement, Int32)

Reads an integer value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int ReadInt (
    string elementName,
    XmlElement parentElement,
    int defaultValue
)
```

Visual Basic Copy Code

```
Public Function ReadInt( _
    elementName As String, _
    parentElement As XmlElement, _
    defaultValue As Integer _
) As Integer
```

Parameters

elementName

parentElement

defaultValue

Return Value

See Also

ReadInt Method Overload List
XmlPersistContext Members

XmlPersistContext.ReadItemReference Method

Overload List

Name	Description
XmlPersistContext.ReadItemReference (String, XmlElement)	Reads a reference to the item with the specified name.
XmlPersistContext.ReadItemReference (XmlElement)	Reads an item reference from the specified element.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadItemReference Method (String, XmlElement)

Reads a reference to the item with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem ReadItemReference (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadItemReference( _
    elementName As String, _
    parentElement As XmlElement _
) As DiagramItem
```

Parameters

elementName

parentElement

Return Value

See Also

ReadItemReference Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadItemReference Method (XmlElement)

Reads an item reference from the specified element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public DiagramItem ReadItemReference (
    XmlElement itemElement
)
```

Visual Basic Copy Code

```
Public Function ReadItemReference( _
    itemElement As XmlElement _
) As DiagramItem
```

Parameters

itemElement

Return Value

See Also

ReadItemReference Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadLong Method

Reads a long value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public long ReadLong (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadLong( _
    elementName As String, _
    parentElement As XmlElement _
) As Long
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadMap Method

Reads a **MapContent** object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public MapContent ReadMap (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadMap( _
    elementName As String, _
    parentElement As XmlElement _
) As MapContent
```

Parameters

elementName
A string specifying the XML element name.

parentElement
An **XmlElement** instance specifying the parent element.

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadNodeConstraints Method

Reads a NodeConstraints object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public NodeConstraints ReadNodeConstraints (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadNodeConstraints( _
    elementName As String, _
    parentElement As XmlElement _
) As NodeConstraints
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadObject Method

Reads an object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object ReadObject (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadObject( _
    elementName As String, _
    parentElement As XmlElement _
) As Object
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadPen Method

Reads a MindFusion.Drawing.Pen object with the specified name.

Overload List

Name	Description
XmlPersistContext.ReadPen (String, XmlElement)	Reads a **MindFusion.Drawing.Pen** object with the specified name.
XmlPersistContext.ReadPen (String, XmlElement, Pen)	Reads a **MindFusion.Drawing.Pen** object with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadPen Method (String, XmlElement)

Reads a MindFusion.Drawing.Pen object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen ReadPen (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadPen( _
    elementName As String, _
    parentElement As XmlElement _
) As Pen
```

Parameters

elementName

parentElement

Return Value

See Also

ReadPen Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadPen Method (String, XmlElement, Pen)

Reads a MindFusion.Drawing.Pen object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Pen ReadPen (
    string elementName,
    XmlElement parentElement,
    Pen defaultPen
)
```

Visual Basic Copy Code

```
Public Function ReadPen( _
    elementName As String, _
    parentElement As XmlElement, _
    defaultPen As Pen _
) As Pen
```

Parameters

elementName

parentElement

defaultPen

Return Value

See Also

ReadPen Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadPointCollection Method

Reads a PointCollection object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public PointCollection ReadPointCollection (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadPointCollection( _
    elementName As String, _
    parentElement As XmlElement _
) As PointCollection
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadPointCoordinates Method

Reads a List<Point> object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public List<Point> ReadPointCoordinates (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadPointCoordinates( _
    elementName As String, _
```

```
    parentElement As XmlElement _
) As List(Of Point)
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadPointF Method

Overload List

Name	Description
XmlPersistContext.ReadPointF (String, XmlElement)	Reads a Point object with the specified name.
XmlPersistContext.ReadPointF (XmlElement)	Reads a Point object from the specified element.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadPointF Method (String, XmlElement)

Reads a Point object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point ReadPointF (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadPointF( _
    elementName As String, _
    parentElement As XmlElement _
) As Point
```

Parameters

elementName

parentElement

Return Value

See Also

ReadPointF Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadPointF Method (XmlElement)

Reads a Point object from the specified element.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Point ReadPointF (
    XmlElement element
)
```

Visual Basic Copy Code

```
Public Function ReadPointF( _
    element As XmlElement _
) As Point
```

Parameters

element

Return Value

See Also

ReadPointF Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadRectangleF Method

Reads a Rect object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Rect ReadRectangleF (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadRectangleF( _
    elementName As String, _
    parentElement As XmlElement _
) As Rect
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadSByte Method

Reads a 8-bit signed integer value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public sbyte ReadSByte (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadSByte( _
    elementName As String, _
    parentElement As XmlElement _
) As SByte
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadShape Method

Reads a Shape object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Shape ReadShape (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadShape( _
    elementName As String, _
    parentElement As XmlElement _
) As Shape
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadShort Method

Overload List

Name	Description
XmlPersistContext.ReadShort (String, XmlElement)	Reads a short value with the specified name.
XmlPersistContext.ReadShort (String, XmlElement, Int16)	Reads a short value with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadShort Method (String, XmlElement)

Reads a short value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public short ReadShort (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadShort( _
    elementName As String, _
    parentElement As XmlElement _
) As Short
```

Parameters

elementName

parentElement

Return Value

See Also

ReadShort Method Overload List
XmlPersistContext Members

XmlPersistContext.ReadShort Method (String, XmlElement, Int16)

Reads a short value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public short ReadShort (
    string elementName,
    XmlElement parentElement,
    short defaultValue
)
```

Visual Basic Copy Code

```
Public Function ReadShort( _
    elementName As String, _
    parentElement As XmlElement, _
    defaultValue As Short _
) As Short
```

Parameters

elementName

parentElement

defaultValue

Return Value

See Also

ReadShort Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadSizeF Method

Reads a Size object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Size ReadSizeF (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadSizeF( _
    elementName As String, _
    parentElement As XmlElement _
) As Size
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadStream Method

Reads a stream with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Stream ReadStream (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadStream( _
    elementName As String, _
    parentElement As XmlElement _
) As Stream
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadStreamOrXml Method

Reads a stream or xml.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object ReadStreamOrXml (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadStreamOrXml( _
    elementName As String, _
    parentElement As XmlElement _
) As Object
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadString Method

Overload List

Name	Description
XmlPersistContext.ReadString (String, XmlElement)	Reads a string value with the specified name.
XmlPersistContext.ReadString (String, XmlElement, String)	Reads a string value with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadString Method (String, XmlElement)

Reads a string value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ReadString (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadString( _
    elementName As String, _
    parentElement As XmlElement _
) As String
```

Parameters

elementName

parentElement

Return Value

See Also

ReadString Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadString Method (String, XmlElement, String)

Reads a string value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ReadString (
    string elementName,
    XmlElement parentElement,
    string defaultValue
)
```

Visual Basic Copy Code

```
Public Function ReadString( _
    elementName As String, _
    parentElement As XmlElement, _
    defaultValue As String _
) As String
```

Parameters

elementName

parentElement

defaultValue

Return Value

See Also

ReadString Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadStringFormat Method

Overload List

Name	Description
XmlPersistContext.ReadStringFormat (String, XmlElement)	Reads a StringFormat object with the specified name.
XmlPersistContext.ReadStringFormat (String, XmlElement, StringFormat)	Reads a StringFormat object with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadStringFormat Method (String, XmlElement)

Reads a StringFormat object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringFormat ReadStringFormat (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadStringFormat( _
    elementName As String, _
    parentElement As XmlElement _
) As StringFormat
```

Parameters

elementName

parentElement

Return Value

See Also

ReadStringFormat Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadStringFormat Method (String, XmlElement, StringFormat)

Reads a StringFormat object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public StringFormat ReadStringFormat (
    string elementName,
    XmlElement parentElement,
    StringFormat defultFormat
)
```

Visual Basic Copy Code

```
Public Function ReadStringFormat( _
    elementName As String, _
    parentElement As XmlElement, _
    defultFormat As StringFormat _
) As StringFormat
```

Parameters

elementName

parentElement

defultFormat

Return Value

See Also

ReadStringFormat Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadStyle Method

Reads a Style object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Style ReadStyle (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadStyle( _
    elementName As String, _
    parentElement As XmlElement _
) As Style
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadSvg Method

Reads a SVG object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public SvgContent ReadSvg (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadSvg( _
    elementName As String, _
```

```
    parentElement As XmlElement _
) As SvgContent
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadTag Method

Reads the specified tag for the specified object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object ReadTag (
    Object item,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadTag( _
    item As Object, _
    elementName As String, _
    parentElement As XmlElement _
) As Object
```

Parameters

item

elementName

parentElement

Return Value

Remarks

Tags are read similarly to arbitrary objects except that the client of the diagram is given a chance to perform custom deserialization through event handling.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadTheme Method

Reads a Theme object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Theme ReadTheme (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadTheme( _
    elementName As String, _
    parentElement As XmlElement _
) As Theme
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadThickness Method

Reads a Thickness object with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Thickness ReadThickness (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadThickness( _
    elementName As String, _
    parentElement As XmlElement _
) As Thickness
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadUInt Method

Reads an unsigned integer value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public uint ReadUInt (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadUInt( _
    elementName As String, _
    parentElement As XmlElement _
) As UInteger
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadULong Method

Reads an unsigned long value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ulong ReadULong (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadULong( _
    elementName As String, _
    parentElement As XmlElement _
) As ULong
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext.ReadUShort Method

Reads an unsigned short value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public ushort ReadUShort (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadUShort( _
    elementName As String, _
    parentElement As XmlElement _
) As UShort
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadVisioContent Method

Reads a Visio content with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public VisioContent ReadVisioContent (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadVisioContent( _
    elementName As String, _
    parentElement As XmlElement _
) As VisioContent
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ReadXmlObject Method

Reads an object using XmlSerializer.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Object ReadXmlObject (
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function ReadXmlObject( _
    elementName As String, _
    parentElement As XmlElement _
) As Object
```

Parameters

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.SelectNode Method

Returns the child node with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement SelectNode (
    XmlElement parentElement,
    string name
)
```

Visual Basic Copy Code

```
Public Function SelectNode( _
    parentElement As XmlElement, _
    name As String _
) As XmlElement
```

Parameters

parentElement
The parent XML element.

name
A string specifying the name of the child node.

Return Value

An XmlElement corresponding to the child node.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.SelectNodes Method

Returns the child nodes with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlNodeList SelectNodes (
    XmlElement parentElement,
    string name
)
```

Visual Basic Copy Code

```
Public Function SelectNodes( _
    parentElement As XmlElement, _
    name As String _
) As XmlNodeList
```

Parameters

parentElement
The parent XML element.

name
A string specifying the name of the child nodes.

Return Value

An XmlNodeList containing the found child nodes.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ToRelativePath Method

Converts the specified absolute path to a path relative to the BaseURI.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string ToRelativePath (
    string path
)
```

Visual Basic Copy Code

```
Public Function ToRelativePath( _
    path As String _
) As String
```

Parameters

path

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteAnchorPattern Method

Writes the specified anchor pattern.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteAnchorPattern (
    AnchorPattern anchorPattern,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteAnchorPattern( _
    anchorPattern As AnchorPattern, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

anchorPattern

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteBool Method

Writes a boolean value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteBool (
    bool boolValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteBool( _
    boolValue As Boolean, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

boolValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteBrush Method

Writes a reference to the specified brush. The brush is registered within the internal hashtable for subsequent serialization.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteBrush (
    Brush brush,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteBrush( _
    brush As Brush, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

brush

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteByte Method

Write a byte value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteByte (
    byte byteValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteByte( _
    byteValue As Byte, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

byteValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteChar Method

Write a char value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteChar (
    char charValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteChar( _
    charValue As Char, _
```

```
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

charValue

elementName

parentElement

Return Value

An instance of the .NET **XmlElement** class representing the newly created XML element.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteColor Method

Writes the specified color.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteColor (
    Color color,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteColor( _
    color As Color, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

color

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteColorArray Method

Writes the specified array.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteColorArray (
    Color[] array,
    string elementName,
    string subElementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteColorArray( _
    array() As Color, _
    elementName As String, _
    subElementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

array

elementName

subElementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteDecimal Method

Write a decimal value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteDecimal (
    decimal decimalValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteDecimal( _
    decimalValue As Decimal, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

decimalValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteDouble Method

Writes a double value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteDouble (
    double doubleValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteDouble( _
    doubleValue As Double, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

doubleValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteEffect Method

Writes a reference to the specified effect. The effect is registered within the internal hashtable for subsequent serialization.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteEffect (
    NodeEffect effect,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteEffect( _
    effect As NodeEffect, _
```

```
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

effect

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteEnum Method

Writes the specified enum value.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteEnum (
    Object enumValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteEnum( _
    enumValue As Object, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

enumValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteFloat Method

Writes a float value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public XmlElement WriteFloat (
    double floatValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic

```
Public Function WriteFloat( _
    floatValue As Double, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

floatValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteFloatArray Method

Writes the specified array.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public XmlElement WriteFloatArray (
    double[] array,
    string elementName,
    string subElementName,
    XmlElement parentElement
)
```

Visual Basic

```
Public Function WriteFloatArray( _
    array() As Double, _
    elementName As String, _
    subElementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

array

elementName

subElementName

parentElement

Return Value

See Also

XmlPersistContext.WriteFont Method

Writes the specified font.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteFont (
    Font font,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteFont( _
    font As Font, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

font

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteGroup Method

Writes the specified group object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteGroup (
    Group group,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteGroup( _
    group As Group, _
    elementName As String, _
```

```
    parentElement As XmlElement _
) As XmlElement
```

Parameters

group

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteHashtable Method

Writes the specified hashtable.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteHashtable (
    Hashtable hashtable,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteHashtable( _
    hashtable As Hashtable, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

hashtable

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteImage Method

Writes a reference to the specified image. The image is registered within the internal hashtable for subsequent serialization.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteImage (
    IImage image,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteImage( _
    image As IImage, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

image

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteInt Method

Writes an integer value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteInt (
    int intValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteInt( _
    intValue As Integer, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

intValue

elementName

parentElement

Return Value

See Also

XmlPersistContext.WriteItemReference Method

Writes a reference to the specified item with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteItemReference (
    DiagramItem item,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteItemReference( _
    item As DiagramItem, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

item

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteLong Method

Writes a long value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteLong (
    long longValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteLong( _
    longValue As Long, _
    elementName As String, _
```

```
    parentElement As XmlElement _
) As XmlElement
```

Parameters

longValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteMap Method

Writes a reference to the specified **MapContent**. The map is registered within the internal hashtable for subsequent serialization.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteMap (
    MapContent map,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteMap( _
    map As MapContent, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

map
The **MapContent** object to be serialized.

elementName
A string specifying the XML element name.

parentElement
An **XmlElement** instance specifying the parent element.

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteObject Method

Writes the specified object under the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteObject (
    Object objectValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteObject( _
    objectValue As Object, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

objectValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WritePen Method

Writes the specified pen.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WritePen (
    Pen pen,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WritePen( _
    pen As Pen, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

pen

elementName

parentElement

Return Value

See Also

XmlPersistContext.WritePointCollection Method

Writes the specified point collection.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WritePointCollection (
    PointCollection points,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WritePointCollection( _
    points As PointCollection, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

points

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WritePointCoordinates Method

Writes the specified List<Point>.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WritePointCoordinates (
    List<Point> points,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WritePointCoordinates( _
    points As List(Of Point), _
    elementName As String, _
```

```
    parentElement As XmlElement _
) As XmlElement
```

Parameters

points

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WritePointF Method

Overload List

Name	Description
XmlPersistContext.WritePointF (Double, Double, String, XmlElement)	Writes the specified Point under the specified name. The x and y components of the point are written as attributes.
XmlPersistContext.WritePointF (Point, String, XmlElement)	Writes a Point with the specified name.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WritePointF Method (Double, Double, String, XmlElement)

Writes the specified Point under the specified name. The x and y components of the point are written as attributes.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WritePointF (
    double x,
    double y,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WritePointF( _
    x As Double, _
    y As Double, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

x

y

elementName

parentElement

Return Value

See Also

WritePointF Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WritePointF Method (Point, String, XmlElement)

Writes a Point with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WritePointF (
    Point point,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WritePointF( _
    point As Point, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

point

elementName

parentElement

Return Value

See Also

WritePointF Method Overload List
XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteRectangleF Method

Writes a Rect with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public XmlElement WriteRectangleF (
    Rect rectangle,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic

```
Public Function WriteRectangleF( _
    rectangle As Rect, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

rectangle

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteSByte Method

Write a 8-bit signed integer value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public XmlElement WriteSByte (
    sbyte byteValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic

```
Public Function WriteSByte( _
    byteValue As SByte, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

byteValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteShape Method

Writes the specified shape with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteShape (
    Shape shape,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteShape( _
    shape As Shape, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

shape

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteShort Method

Writes a short value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteShort (
    short shortValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteShort( _
    shortValue As Short, _
    elementName As String, _
```

```
    parentElement As XmlElement _
) As XmlElement
```

Parameters

shortValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteSizeF Method

Writes a Size with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteSizeF (
    Size size,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteSizeF( _
    size As Size, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

size

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteStream Method

Writes the specified stream.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteStream (
    Stream stream,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteStream( _
    stream As Stream, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

stream

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteString Method

Writes a string value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteString (
    string stringValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteString( _
    stringValue As String, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

stringValue

elementName

parentElement

Return Value

See Also

XmlPersistContext.WriteStringFormat Method

Writes the specified StringFormat object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteStringFormat (
    StringFormat stringFormat,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteStringFormat( _
    stringFormat As StringFormat, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

stringFormat

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteStyle Method

Writes a reference to the specified style. The style is registered within the internal hashtable for subsequent serialization.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteStyle (
    Style style,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteStyle( _
    style As Style, _
    elementName As String, _
```

```
    parentElement As XmlElement _
) As XmlElement
```

Parameters

style

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteSvg Method

Writes a reference to the specified SVG. The SVG is registered within the internal hashtable for subsequent serialization.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteSvg (
    SvgContent svg,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteSvg( _
    svg As SvgContent, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

svg

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteTag Method

Writes the specified tag of the specified object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteTag (
    Object item,
    Object tagValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteTag( _
    item As Object, _
    tagValue As Object, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

item

tagValue

elementName

parentElement

Return Value

Remarks

Tags are written similarly to arbitrary objects except that the client of the diagram is given a chance to perform custom serialization through event handling.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteTheme Method

Writes the specified Theme object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteTheme (
    Theme theme,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteTheme( _
    theme As Theme, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

theme

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteThickness Method

Writes a Thickness with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteThickness (
    Thickness thickness,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteThickness( _
    thickness As Thickness, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

thickness

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteTraits Method

Writes the specified traits hashtable.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteTraits (
    Hashtable traits,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic

```
Public Function WriteTraits( _
    traits As Hashtable, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

traits

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteUInt Method

Writes an unsigned integer value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C#

```
public XmlElement WriteUInt (
    uint uintValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic

```
Public Function WriteUInt( _
    uintValue As UInteger, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

uintValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteULong Method

Writes an unsigned long value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteULong (
    ulong ulongValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteULong( _
    ulongValue As ULong, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

ulongValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteUShort Method

Writes an unsigned short value with the specified name.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteUShort (
    ushort ushortValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteUShort( _
    ushortValue As UShort, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

ushortValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteVisioContent Method

Writes a reference to the specified Visio content. The content is registered within the internal hashtable for subsequent serialization.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteVisioContent (
    VisioContent content,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteVisioContent( _
    content As VisioContent, _
    elementName As String, _
    parentElement As XmlElement _
) As XmlElement
```

Parameters

content

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.WriteXmlObject Method

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlElement WriteXmlObject (
    Object objectValue,
    string elementName,
    XmlElement parentElement
)
```

Visual Basic Copy Code

```
Public Function WriteXmlObject( _
    objectValue As Object, _
    elementName As String, _
```

```
    parentElement As XmlElement _
) As XmlElement
```

Parameters

objectValue

elementName

parentElement

Return Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext Properties

XmlPersistContext.BaseURI Property

Gets or sets the base URI of the XML document being loaded or saved. When not specified, the BaseURI of the underlying XmlDocument is used.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public string BaseURI { get; set; }
```

Visual Basic Copy Code

```
Public Property BaseURI As String
```

Property Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.ClassVersion Property

Gets the class version of the currently deserialized item.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int ClassVersion { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ClassVersion As Integer
```

Property Value

An integer value specifying the class version.

Remarks

This property is only useful when deserializing items.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.Diagram Property

Gets the underlying diagram object.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public Diagram Diagram { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Diagram As Diagram
```

Property Value

An instance of the Diagram class.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.FileVersion Property

Gets the format revision number for the file being currently serialized.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public int FileVersion { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property FileVersion As Integer
```

Property Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.IncludeUnalteredProperties Property

Gets or sets a value specifies whether properties that have their default value should be serialized to XML.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public bool IncludeUnalteredProperties { get; set; }
```

Visual Basic Copy Code

```
Public Property IncludeUnalteredProperties As Boolean
```

Property Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.XmlDocument Property

Gets the underlying XML document.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlDocument XmlDocument { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property XmlDocument As XmlDocument
```

Property Value

An instance of the .NET **XmlDocument** class.

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

XmlPersistContext.XmlOptions Property

An XmlOptions object providing properties for customizing the XML serialization process.

Namespace: MindFusion.Diagramming
Assembly: MindFusion.Diagramming

Syntax

C# Copy Code

```
public XmlOptions XmlOptions { get; set; }
```

Visual Basic

```
Public Property XmlOptions As XmlOptions
```

Property Value

See Also

XmlPersistContext Members
XmlPersistContext Class
MindFusion.Diagramming Namespace

MindFusion.Diagramming.Export Namespace

AutoScale Enumeration

Specifies the auto-scale mode when exporting to a PDF document.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C#

```
public enum AutoScale
```

Visual Basic

```
Public Enum AutoScale
```

Members

	Member name	Description
	None	No automatic scaling.
	FitToPage	Scales the chart (aspect ratio is preserved) to fit the page bounds if the document is too large.
	FitToPageNoUpScale	Scales down the chart (aspect ratio is preserved) to fit the page bounds if the document is too large. Otherwise chart's size is left untouched.
	FitToWidth	Scales down the chart (aspect ratio is preserved) to make sure that the width of the chart is less than the width of the PDF page.

See Also

MindFusion.Diagramming.Export Namespace

DxfExporter Class

Exports diagrams to AutoDesk Drawing Exchange Format (DXF) files.

Syntax

C#

```
public class DxfExporter
```

Visual Basic

```
Public Class DxfExporter
```

Remarks

In order to use this component, add a reference to the *MindFusion.Diagramming.Export.Dxf.dll* to your project.

AutoCAD DXF (Drawing Exchange Format) is a file format developed by Autodesk to enable interoperability between AutoCAD and other programs. Now virtually every CAD application and most graphics programs can import DXF files.

To create a DXF file from a MvcDiagram diagram, create a **DxfExporter** instance and call its Export method, specifying a Diagram object and a file path as arguments.

Raster images can be used in DXF by referencing external image files. To avoid creating extraneous files, **DxfExporter** does not export images by default. To export the images displayed in diagram elements, enable the ExportImages property.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Export.DxfExporter

See Also

DxfExporter Members
MindFusion.Diagramming.Export Namespace

DxfExporter Members

Exports diagrams to AutoDesk Drawing Exchange Format (DXF) files.

The following tables list the members exposed by the DxfExporter type.

Public Constructors

	Name	Description
	DxfExporter	Initializes a new instance of the **DxfExporter** class.

Public Properties

	Name	Description
	ExportImages	Gets or sets a value indicating whether the images that are displayed inside diagram nodes should be exported.
	ExportTextAsMultiline	Gets or sets a value indicating whether item labels should be exported to DXF MTEXT entities.

Public Methods

	Name	Description
	Export	Creates a DXF file representing the specified MvcDiagram diagram.

See Also

DxfExporter Class
MindFusion.Diagramming.Export Namespace

DxfExporter Constructor

Initializes a new instance of the DxfExporter class.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Dxf

Syntax

C# Copy Code

```
public DxfExporter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DxfExporter Class
MindFusion.Diagramming.Export Namespace

DxfExporter Methods

DxfExporter.Export Method

Creates a DXF file representing the specified MvcDiagram diagram.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Dxf

Syntax

C# Copy Code

```
public void Export (
    Diagram diagram,
    string filePath
)
```

Visual Basic Copy Code

```
Public Sub Export ( _
    diagram As Diagram, _
    filePath As String _
)
```

Parameters

diagram
A reference to the diagram to be exported.

filePath
The full path to the .dxf file.

Remarks

Each MvcDiagram node or link is exported as a group of graphic primitives, such as lines and arcs, defined as DXF entities. DxfExporter uses the entities defined in version 14 of the DXF format.

See Also

DxfExporter Members
DxfExporter Class
MindFusion.Diagramming.Export Namespace

DxfExporter Properties

DxfExporter.ExportImages Property

Gets or sets a value indicating whether the images that are displayed inside diagram nodes should be exported.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Dxf

Syntax

C# Copy Code

```
public bool ExportImages { get; set; }
```

Visual Basic Copy Code

```
Public Property ExportImages As Boolean
```

Property Value

true to export images; otherwise, **false**. The default is **false**.

Remarks

If this property is enabled, images displayed in MvcDiagram nodes are saved into external files and references to these files are placed inside the generated DXF code. The image files have the same base name as the specified DXF name, followed by an underscore symbol and a sequential number.

See Also

DxfExporter Members
DxfExporter Class
MindFusion.Diagramming.Export Namespace

DxfExporter.ExportTextAsMultiline Property

Gets or sets a value indicating whether item labels should be exported to DXF MTEXT entities.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Dxf

Syntax

C# Copy Code

```
public bool ExportTextAsMultiline { get; set; }
```

Visual Basic Copy Code

```
Public Property ExportTextAsMultiline As Boolean
```

Property Value

true to create multiple-line text entities, otherwise, **false**. The default is **true**.

Remarks

Some applications do not support the multiple-line text DXF entity - MTEXT. If you notice that the application you target does not display the labels of diagram items, set this property to **false**.

See Also

DxfExporter Members
DxfExporter Class
MindFusion.Diagramming.Export Namespace

ExcelExporter Class

Exports MindFusion.Diagramming diagrams to Excel Open XML Format (XLSX) files.

Syntax

C# Copy Code

```
public class ExcelExporter
```

Visual Basic Copy Code

```
Public Class ExcelExporter
```

Remarks

Diagrams are exported as Excel AutoShapes drawings. The Export overload that takes a DiagramDocument parameter creates a sheet for each DiagramPage in the document.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Export.ExcelExporter

See Also

ExcelExporter Members
MindFusion.Diagramming.Export Namespace

ExcelExporter Members

Exports MindFusion.Diagramming diagrams to Excel Open XML Format (XLSX) files.

The following tables list the members exposed by the ExcelExporter type.

Public Constructors

	Name	Description
	ExcelExporter	Initializes a new instance of the **ExcelExporter** class.

Public Methods

	Name	Description
	Export	Overloaded.

See Also

ExcelExporter Class
MindFusion.Diagramming.Export Namespace

ExcelExporter Constructor

Initializes a new instance of the ExcelExporter class.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Excel

Syntax

C# Copy Code

```
public ExcelExporter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ExcelExporter Methods

ExcelExporter.Export Method

Overload List

Name	Description
ExcelExporter.Export (Diagram, Stream)	Creates an XLSX file representing the specified diagram.
ExcelExporter.Export (Diagram, String)	Creates an XLSX file representing the specified diagram.
ExcelExporter.Export (DiagramDocument, Stream)	Creates an XLSX file representing the specified DiagramDocument.
ExcelExporter.Export (DiagramDocument, String)	Creates an XLSX file representing the specified **DiagramDocument**.

See Also

ExcelExporter Members
ExcelExporter Class
MindFusion.Diagramming.Export Namespace

ExcelExporter.Export Method (Diagram, Stream)

Creates an XLSX file representing the specified diagram.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Excel

Syntax

C# Copy Code

```
public void Export (
    Diagram diagram,
    Stream stream
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagram As Diagram, _
    stream As Stream _
)
```

Parameters

diagram
A reference to the Diagram that will be exported.

stream
Output **System.IO.Stream** to which to write the Excel data.

See Also

Export Method Overload List
ExcelExporter Members
ExcelExporter Class
MindFusion.Diagramming.Export Namespace

ExcelExporter.Export Method (Diagram, String)

Creates an XLSX file representing the specified diagram.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Excel

Syntax

C# Copy Code

```
public void Export (
    Diagram diagram,
    string filePath
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagram As Diagram, _
    filePath As String _
)
```

Parameters

diagram
A reference to the Diagram that will be exported.

filePath
The full path to the XLSX file.

See Also

Export Method Overload List
ExcelExporter Members
ExcelExporter Class
MindFusion.Diagramming.Export Namespace

ExcelExporter.Export Method (DiagramDocument, Stream)

Creates an XLSX file representing the specified DiagramDocument.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Excel

Syntax

C# Copy Code

```
public void Export (
    DiagramDocument document,
    Stream stream
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    document As DiagramDocument, _
    stream As Stream _
)
```

Parameters

document
A reference to the DiagramDocument that will be exported.

stream
Output **System.IO.Stream** to which to write the Excel data.

See Also

Export Method Overload List
ExcelExporter Members
ExcelExporter Class
MindFusion.Diagramming.Export Namespace

ExcelExporter.Export Method (DiagramDocument, String)

Creates an XLSX file representing the specified DiagramDocument.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Excel

Syntax

C# Copy Code

```
public void Export (
    DiagramDocument document,
    string filePath
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    document As DiagramDocument, _
    filePath As String _
)
```

Parameters

document
A reference to the DiagramDocument that will be exported.

filePath
The full path to the XLSX file.

See Also

Export Method Overload List
ExcelExporter Members
ExcelExporter Class
MindFusion.Diagramming.Export Namespace

ExportLanes Enumeration

Defines values that indicate how the LaneGrid should be exported to Visio.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public enum ExportLanes
```

Visual Basic Copy Code

```
Public Enum ExportLanes
```

Members

	Member name	Description

	None	Do not export the swimlane grid.
	Horizontal	Export the grid as horizontal functional band shapes.
	Vertical	Export the grid as vertical functional band shapes.

See Also

MindFusion.Diagramming.Export Namespace

PdfExporter Class

Exports diagrams to Adobe Portable Document Format (PDF) files.

Syntax

C# Copy Code

```
public class PdfExporter
```

Visual Basic Copy Code

```
Public Class PdfExporter
```

Remarks

In order to use this component, add references to the *MindFusion.Diagramming.Export.Pdf.dll* and *MindFusion.Pdf.dll* assemblies to your project.

The Adobe Portable Document Format (PDF) is the most popular open format for representing textual and graphic information in resolution independent manner. In addition, PDF is a multiplatform format, viewable and printable on all popular operating systems. These features, along with its ability to describe complex documents that use a variety of graphics, images and fonts, make PDF a good choice when it comes to distributing your flowcharts and diagrams to others.

The **PdfExporter** class lets you create PDF files that represent MvcDiagram diagrams. In order to do that, create a **PdfExporter** instance and call its Export method. If you need to set the page size and orientation, use the PageSize, and PageOrientation properties. Margins specifies the page margins for all pages in the document. The AutoScale property lets you scale the diagram so it fits in one or more PDF pages.

Pages can display optional headers and footers. Set the HeaderFormat and HeaderFont properties to customize headers, and the FooterFormat and FooterFont properties to customize footers.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Export.PdfExporter

See Also

PdfExporter Members
MindFusion.Diagramming.Export Namespace

PdfExporter Members

Exports diagrams to Adobe Portable Document Format (PDF) files.

The following tables list the members exposed by the PdfExporter type.

Public Constructors

	Name	Description
	PdfExporter	Initializes a new instance of the **PdfExporter** class.

Public Properties

	Name	Description
	AutoDetectEncoding	Specifies whether character encoding should be detected automatically.
	AutoScale	Gets or sets a value indicating whether the diagram should be scaled to fit one or more pages of the PDF document.
	DefaultEncoding	Specifies what encoding to use if the encoding for a string cannot be recognized automatically.
	DisableTextClipping	A work-around for Adobe Reader V9 issue when printing to Postscript printers.
	EnableBackground	Specifies whether to fill the background with color.
	EnableGrid	Gets or sets a value indicating whether the alignment grid is rendered.
	FontCachePath	Specifies the path to a file that contains font metrics records.
	FooterFont	Gets or sets the font used to draw page footers.
	FooterFormat	Gets or sets a string specifying the content of page footers.
	HeaderFont	Gets or sets the font used to draw page headers.
	HeaderFormat	Gets or sets a string specifying the content of page headers.
	HyperLinkStyle	Specifies what kind of hyperlinks should be generated for the HyperLink property of items.
	InvisibleItems	Gets or sets a value indicating whether invisible items should be exported.
	JpegImageEncoding	Specifies whether images embedded in PDF should be encoded in Jpeg format.
	JpegImageQuality	Specifies the image quality of Jpeg images.
	Margins	Gets or sets the margins for the PDF pages.
	PageOrientation	Gets or sets the orientation of the PDF pages.
	PageSize	Gets or sets the size of the PDF pages.
	RenderOptions	Gets a RenderOptions instance that provides various settings for configuring the appearance of exported diagrams.
	Scale	Gets or sets the scale of the PDF output relatively to the original on-screen size of the diagram.
	SuppressUnmanagedCode	Suppress usage of unmanaged code.

Public Methods

	Name	Description
	Export	Overloaded.

See Also

PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter Constructor

Initializes a new instance of the PdfExporter class.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public PdfExporter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter Methods

PdfExporter.Export Method

Overload List

Name	Description
PdfExporter.Export (Diagram, Stream)	Creates a PDF file representing the specified MindFusion.Diagramming diagram.
PdfExporter.Export (Diagram, String)	Creates a PDF file representing the specified MvcDiagram diagram.
PdfExporter.Export (DiagramDocument, Stream)	Creates a PDF file representing the specified DiagramDocument.
PdfExporter.Export (DiagramDocument, String)	Creates a PDF file representing the specified DiagramDocument.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.Export Method (Diagram, Stream)

Creates a PDF file representing the specified MindFusion.Diagramming diagram.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public void Export (
    Diagram diagram,
    Stream stream
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagram As Diagram, _
    stream As Stream _
)
```

Parameters

diagram
A reference to the diagram that will be exported.

stream
Output **System.IO.Stream** to which to write the PDF data.

See Also

Export Method Overload List
PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.Export Method (Diagram, String)

Creates a PDF file representing the specified MvcDiagram diagram.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public void Export (
    Diagram diagram,
    string filePath
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagram As Diagram, _
    filePath As String _
)
```

Parameters

diagram
A reference to the diagram that will be exported.

filePath
The full path to the PDF file.

Remarks

By default, PdfExporter preserves the original scale of the diagram. If Bounds can not fit in a single page of size PageSize, the exported diagram is divided into multiple pages. The AutoScale property allows scaling the diagram so that it exactly fits in one or more pages.

See Also

Export Method Overload List
PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.Export Method (DiagramDocument, Stream)

Creates a PDF file representing the specified DiagramDocument.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public void Export (
    DiagramDocument diagramDoc,
    Stream stream
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagramDoc As DiagramDocument, _
    stream As Stream _
)
```

Parameters

diagramDoc
A reference to the DiagramDocument that will be exported.

stream
Output **System.IO.Stream** to which to write the PDF data.

Remarks

If PageSize is set to **DiagramSize**, the number of pages in the output PDF is equal to DiagramDocument.Pages.Count and each PDF page is as large as the respective DiagramPage.Bounds. Otherwise, all diagram pages are paginated using the specified PDF page size (A4 etc).

See Also

Export Method Overload List
PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.Export Method (DiagramDocument, String)

Creates a PDF file representing the specified DiagramDocument.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public void Export (
    DiagramDocument diagramDoc,
    string filePath
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagramDoc As DiagramDocument, _
    filePath As String _
)
```

Parameters

diagramDoc
A reference to the DiagramDocument that will be exported.

filePath
The full path to the PDF file.

Remarks

If PageSize is set to DiagramSize, the number of pages in the output PDF is equal to DiagramDocument.Pages.Count and each PDF page is as large as the respective DiagramPage.Bounds. Otherwise, all diagram pages are paginated using the specified PDF page size (A4 etc).

See Also

Export Method Overload List
PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter Properties

PdfExporter.AutoDetectEncoding Property

Specifies whether character encoding should be detected automatically.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public bool AutoDetectEncoding { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoDetectEncoding As Boolean
```

Property Value

true to detect character encoding automatically, otherwise **false**. The default value is **true**.

Remarks

If this property is disabled, the PDF encoding is always set to the DefaultEncoding value.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.AutoScale Property

Gets or sets a value indicating whether the diagram should be scaled to fit one or more pages of the PDF document.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public AutoScale AutoScale { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoScale As AutoScale
```

Property Value

A member of the AutoScale enumeration. The default is **None**.

Remarks

By default, the original scale of the diagram is preserved and the diagram is divided into multiple pages if needed. This property lets you scale the diagram so that it exactly fits in one or more pages.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace
PageSize Property
Scale Property

PdfExporter.DefaultEncoding Property

Specifies what encoding to use if the encoding for a string cannot be recognized automatically.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public Encoding DefaultEncoding { get; set; }
```

Visual Basic Copy Code

```
Public Property DefaultEncoding As Encoding
```

Property Value

An instance of the **System.Text.Encoding** class.

Remarks

PDF text encoding is set to this value if automatic encoding detection fails or when any of the AutoDetectEncoding and SuppressUnmanagedCode properties are disabled.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.DisableTextClipping Property

A work-around for Adobe Reader V9 issue when printing to Postscript printers.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public bool DisableTextClipping { get; set; }
```

Visual Basic Copy Code

```
Public Property DisableTextClipping As Boolean
```

Property Value

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.EnableBackground Property

Specifies whether to fill the background with color.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public bool EnableBackground { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableBackground As Boolean
```

Property Value

true to paint the page background with the diagram's BackBrush, otherwise **false**. The default value is **true**.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.EnableGrid Property

Gets or sets a value indicating whether the alignment grid is rendered.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public bool EnableGrid { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableGrid As Boolean
```

Property Value

true to render the alignment grid, if visible; otherwise, **false**.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.FontCachePath Property

Specifies the path to a file that contains font metrics records.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public string FontCachePath { get; set; }
```

Visual Basic

```
Public Property FontCachePath As String
```

Property Value

A string specifying the full path to *pdflib-fontrecords.xml*.

Remarks

This property must be set if SuppressUnmanagedCode is disabled.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.FooterFont Property

Gets or sets the font used to draw page footers.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C#

```
public Font FooterFont { get; set; }
```

Visual Basic

```
Public Property FooterFont As Font
```

Property Value

A Font instance.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.FooterFormat Property

Gets or sets a string specifying the content of page footers.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C#

```
public string FooterFormat { get; set; }
```

Visual Basic

```
Public Property FooterFormat As String
```

Property Value

A string specifying the footer content. The default value is **null**.

Remarks

A %P placeholder in this string is replaced with the current page number.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.HeaderFont Property

Gets or sets the font used to draw page headers.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public Font HeaderFont { get; set; }
```

Visual Basic Copy Code

```
Public Property HeaderFont As Font
```

Property Value

A Font instance.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.HeaderFormat Property

Gets or sets a string specifying the content of page headers.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public string HeaderFormat { get; set; }
```

Visual Basic Copy Code

```
Public Property HeaderFormat As String
```

Property Value

A string specifying the header content. The default value is **null**.

Remarks

A %P placeholder in this string is replaced with the current page number.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.HyperLinkStyle Property

Specifies what kind of hyperlinks should be generated for the HyperLink property of items.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public HyperLinkStyle HyperLinkStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property HyperLinkStyle As HyperLinkStyle
```

Property Value

A member of the HyperLinkStyle enumeration. The default is **Text**.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.InvisibleItems Property

Gets or sets a value indicating whether invisible items should be exported.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public bool InvisibleItems { get; set; }
```

Visual Basic Copy Code

```
Public Property InvisibleItems As Boolean
```

Property Value

true if invisible items should be displayed in the PDF; otherwise, **false**. The default is **false**.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.JpegImageEncoding Property

Specifies whether images embedded in PDF should be encoded in Jpeg format.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public bool JpegImageEncoding { get; set; }
```

Visual Basic

```
Public Property JpegImageEncoding As Boolean
```

Property Value

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.JpegImageQuality Property

Specifies the image quality of Jpeg images.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C#

```
public int JpegImageQuality { get; set; }
```

Visual Basic

```
Public Property JpegImageQuality As Integer
```

Property Value

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.Margins Property

Gets or sets the margins for the PDF pages.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C#

```
public Thickness Margins { get; set; }
```

Visual Basic

```
Public Property Margins As Thickness
```

Property Value

A Thickness instance that represents the margins in hundredths of an inch. The default is half-inch margins on all sides.

Example

The following C# code creates a PDF file with one-inch page margins:

C#

```
var pdfExporter = new PdfExporter();
pdfExporter.Margins = new Thickness(100, 100, 100, 100);
pdfExporter.Export(diagram, stream);
```

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace
PageSize Property

PdfExporter.PageOrientation Property

Gets or sets the orientation of the PDF pages.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public PageOrientation PageOrientation { get; set; }
```

Visual Basic Copy Code

```
Public Property PageOrientation As PageOrientation
```

Property Value

A member of the **PageOrientation** enumeration. The default is **Portrait**. The members of the **PageOrientation** enumeration are: **Landscape**, **Portrait** or **Auto**.

Remarks

If the **Auto** value is set, PdfExporter will choose the appropriate orientation according to the proportion of width to height of the diagram.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace
PageSize Property

PdfExporter.PageSize Property

Gets or sets the size of the PDF pages.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public PageSize PageSize { get; set; }
```

Visual Basic Copy Code

```
Public Property PageSize As PageSize
```

Property Value

A member of the **PageSize** enumeration. The default is **A4**.

Remarks

The diagram is divided into multiple pages of the specified size, as long as AutoScale is not set to FitToPage.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace
PageOrientation Property
Margins Property

PdfExporter.RenderOptions Property

Gets a RenderOptions instance that provides various settings for configuring the appearance of exported diagrams.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public RenderOptions RenderOptions { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property RenderOptions As RenderOptions
```

Property Value

A RenderOptions reference.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.Scale Property

Gets or sets the scale of the PDF output relatively to the original on-screen size of the diagram.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public float Scale { get; set; }
```

Visual Basic Copy Code

```
Public Property Scale As Single
```

Property Value

A float value specifying the scale as percentage of the original size. The default is 100.

Remarks

This value is used only if AutoScale is disabled.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

PdfExporter.SuppressUnmanagedCode Property

Suppress usage of unmanaged code.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Pdf

Syntax

C# Copy Code

```
public bool SuppressUnmanagedCode { get; set; }
```

Visual Basic Copy Code

```
Public Property SuppressUnmanagedCode As Boolean
```

Property Value

true to prevent the component from using unmanaged code; **false** to allow it. The default value is **false**.

Remarks

Unmanaged code is not allowed to run when the ASP.NET security level is set to Medium Trust. However PdfExporter relies on native Win32 calls to get some precise font metrics which are otherwise inaccessible through the .NET base class library. To work around this, the component now comes with a "pdflib-fontrecords.xml" file that contains cached information about fonts. The FontCachePath property must point to the location of this file when **SuppressUnmanagedCode** is enabled.

Unmanaged code is also used to detect character encoding automatically. When it is suppressed, PDF encoding is always set to the value of DefaultEncoding.

See Also

PdfExporter Members
PdfExporter Class
MindFusion.Diagramming.Export Namespace

SvgExporter Class

Creates SVG (Scalable Vector Graphics) drawings from the content of Diagram objects.

Syntax

C# Copy Code

```
public class SvgExporter
```

Visual Basic Copy Code

```
Public Class SvgExporter
```

Remarks

In order to use this component, add references to the *MindFusion.Diagramming.Export.Svg.dll* and *MindFusion.Svg.dll* assemblies to your project.

Scalable Vector Graphics is an XML-based graphics language that describes images with vector shapes, text, and embedded raster graphics. SVG drawings can be displayed in web pages, or edited / imported by vector graphics editing applications.

In order to create an SVG file from a MvcDiagram diagram, create an **SvgExporter** instance and call its Export method, passing a Diagram object as parameter.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.Export.SvgExporter

See Also

SvgExporter Members

Creates SVG (Scalable Vector Graphics) drawings from the content of Diagram objects.

The following tables list the members exposed by the SvgExporter type.

Public Constructors

	Name	Description
	SvgExporter	Initializes a new instance of the **SvgExporter** class.

Public Properties

	Name	Description
	ExternalImages	Specifies whether images displayed by diagram items should be exported to external files.
	GZipCompression	Specifies whether to create GZip compressed SVG.
	HyperLinkStyle	Specifies what kind of hyperlinks should be generated for the HyperLink property of items.
	HyperLinkTarget	Specify the value of the **target** attribute generated for the HyperLink property of items.
	RenderOptions	Gets a RenderOptions instance that provides various settings for configuring the appearance of exported diagrams.

Public Methods

	Name	Description
	Export	Overloaded.

See Also

SvgExporter Class
MindFusion.Diagramming.Export Namespace

SvgExporter Constructor

Initializes a new instance of the SvgExporter class.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Svg

Syntax

C# Copy Code

```
public SvgExporter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SvgExporter Methods

SvgExporter.Export Method

Overload List

Name	Description
SvgExporter.Export (Diagram, Stream)	Saves SVG markup representing the specified diagram to a stream.
SvgExporter.Export (Diagram, String)	Creates an SVG file representing the specified diagram.

See Also

SvgExporter Members
SvgExporter Class
MindFusion.Diagramming.Export Namespace

SvgExporter.Export Method (Diagram, Stream)

Saves SVG markup representing the specified diagram to a stream.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Svg

Syntax

C# Copy Code

```
public void Export (
    Diagram diagram,
    Stream stream
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagram As Diagram, _
    stream As Stream _
)
```

Parameters

diagram
A reference to the diagram to be exported.

stream
Output **System.IO.Stream** to which to write the SVG data.

See Also

Export Method Overload List
SvgExporter Members
SvgExporter Class
MindFusion.Diagramming.Export Namespace

SvgExporter.Export Method (Diagram, String)

Creates an SVG file representing the specified diagram.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Svg

Syntax

C# Copy Code

```
public void Export (
    Diagram diagram,
    string filePath
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagram As Diagram, _
    filePath As String _
)
```

Parameters

diagram
A reference to the diagram to be exported.

filePath
A full path to the .svg file.

See Also

Export Method Overload List
SvgExporter Members
SvgExporter Class
MindFusion.Diagramming.Export Namespace

SvgExporter Properties

SvgExporter.ExternalImages Property

Specifies whether images displayed by diagram items should be exported to external files.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Svg

Syntax

C# Copy Code

```
public bool ExternalImages { get; set; }
```

Visual Basic Copy Code

```
Public Property ExternalImages As Boolean
```

Property Value

true to export images to external files; **false** to keep them as inline binary data in the SVG file. The default is **false**.

See Also

SvgExporter Members
SvgExporter Class
MindFusion.Diagramming.Export Namespace

SvgExporter.GZipCompression Property

Specifies whether to create GZip compressed SVG.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Svg

Syntax

C# Copy Code

```
public bool GZipCompression { get; set; }
```

Visual Basic Copy Code

```
Public Property GZipCompression As Boolean
```

Property Value

See Also

SvgExporter Members
SvgExporter Class
MindFusion.Diagramming.Export Namespace

SvgExporter.HyperLinkStyle Property

Specifies what kind of hyperlinks should be generated for the HyperLink property of items.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Svg

Syntax

C# Copy Code

```
public HyperLinkStyle HyperLinkStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property HyperLinkStyle As HyperLinkStyle
```

Property Value

A member of the HyperLinkStyle enumeration. The default is **Text**.

See Also

SvgExporter Members
SvgExporter Class
MindFusion.Diagramming.Export Namespace

SvgExporter.HyperLinkTarget Property

Specify the value of the **target** attribute generated for the HyperLink property of items.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Svg

Syntax

C# Copy Code

```
public string HyperLinkTarget { get; set; }
```

Visual Basic Copy Code

```
Public Property HyperLinkTarget As String
```

Property Value

The default is "_self".

See Also

SvgExporter Members
SvgExporter Class
MindFusion.Diagramming.Export Namespace

SvgExporter.RenderOptions Property

Gets a RenderOptions instance that provides various settings for configuring the appearance of exported diagrams.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Svg

Syntax

C# Copy Code

```
public RenderOptions RenderOptions { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property RenderOptions As RenderOptions
```

Property Value

A RenderOptions reference.

See Also

SvgExporter Members
SvgExporter Class
MindFusion.Diagramming.Export Namespace

Visio2013Exporter Class

Exports the contents of Diagram or DiagramDocument objects to Visio 2013 XML drawing files (*.vsdx).

Syntax

C# Copy Code

```
public class Visio2013Exporter
```

Visual Basic Copy Code

```
Public Class Visio2013Exporter
```

Remarks

Add a reference to the *MindFusion.Diagramming.Export.Visio.dll* to your project in order to use this component.

The component allows exporting Visio 2003 XML Drawing files from MindFusion.Diagramming. ShapeNode objects are mapped to their closest Visio counterparts and DiagramLink instances are exported as Visio connectors of a similar style. In addition, most of the attributes of diagram items such as text, text formatting, fill and line colors and styles are preserved.

To function properly, the exporter object requires the *Visio2013.vxt* file to be present in the application's directory, or its location specified in TemplatePath. This file contains an XML template data used by the component to generate Visio documents.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.Export.Visio2013Exporter

See Also

Visio2013Exporter Members
MindFusion.Diagramming.Export Namespace

Visio2013Exporter Members

Exports the contents of Diagram or DiagramDocument objects to Visio 2013 XML drawing files (*.vsdx).

The following tables list the members exposed by the Visio2013Exporter type.

Public Constructors

	Name	Description
	Visio2013Exporter	Initializes a new instance of the **Visio2013Exporter** class.

Public Properties

	Name	Description
	TemplatePath	Gets or sets the full path to the 'Visio2013.vxt' template file.

Public Methods

	Name	Description
	Export	Overloaded.

See Also

Visio2013Exporter Class
MindFusion.Diagramming.Export Namespace

Visio2013Exporter Constructor

Initializes a new instance of the Visio2013Exporter class.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public Visio2013Exporter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

Visio2013Exporter Class
MindFusion.Diagramming.Export Namespace

Visio2013Exporter Methods

Visio2013Exporter.Export Method

Overload List

Name	Description
Visio2013Exporter.Export (Diagram, Stream)	Creates a Visio .vsdx file representing the specified Diagram.
Visio2013Exporter.Export (Diagram, String)	Creates a Visio .vsdx file representing the specified Diagram.
Visio2013Exporter.Export (DiagramDocument, Stream)	Creates a Visio .vsdx file representing the specified DiagramDocument.
Visio2013Exporter.Export (DiagramDocument, String)	Creates a Visio .vsdx file representing the specified DiagramDocument.

See Also

Visio2013Exporter Members
Visio2013Exporter Class
MindFusion.Diagramming.Export Namespace

Visio2013Exporter.Export Method (Diagram, Stream)

Creates a Visio .vsdx file representing the specified Diagram.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public void Export (
    Diagram diagram,
    Stream output
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagram As Diagram, _
    output As Stream _
)
```

Parameters

diagram

output

See Also

Export Method Overload List
Visio2013Exporter Members
Visio2013Exporter Class
MindFusion.Diagramming.Export Namespace

Visio2013Exporter.Export Method (Diagram, String)

Creates a Visio .vsdx file representing the specified Diagram.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public void Export (
    Diagram diagram,
    string filePath
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagram As Diagram, _
    filePath As String _
)
```

Parameters

diagram
A Diagram instance.

filePath
The full path to the exported Visio .vsdx file.

Remarks

The exported document contains a single page, as big as the scrollable diagram area set in Bounds.

To run properly, this method requires the *Visio2013.vxt* file to be present in the application's directory, or its full path to be assigned to TemplatePath. That file contains an XML template data used by the component to construct Visio documents.

See Also

Export Method Overload List
Visio2013Exporter Members
Visio2013Exporter Class
MindFusion.Diagramming.Export Namespace

Visio2013Exporter.Export Method (DiagramDocument, Stream)

Creates a Visio .vsdx file representing the specified DiagramDocument.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public void Export (
    DiagramDocument doc,
    Stream output
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    doc As DiagramDocument, _
    output As Stream _
)
```

Parameters

doc

output

See Also

Export Method Overload List
Visio2013Exporter Members
Visio2013Exporter Class
MindFusion.Diagramming.Export Namespace

Visio2013Exporter.Export Method (DiagramDocument, String)

Creates a Visio .vsdx file representing the specified DiagramDocument.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public void Export (
    DiagramDocument doc,
    string filePath
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    doc As DiagramDocument, _
    filePath As String _
)
```

Parameters

doc
A DiagramDocument instance.

filePath
The full path to the exported Visio .vsdx file.

Remarks

Each DiagramPage in the document is exported to an individual Visio page. The title of the Visio page is set to DiagramPage.Title.

See Also

Export Method Overload List
Visio2013Exporter Members
Visio2013Exporter Class
MindFusion.Diagramming.Export Namespace

Visio2013Exporter Properties

Visio2013Exporter.TemplatePath Property

Gets or sets the full path to the 'Visio2013.vxt' template file.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public string TemplatePath { get; set; }
```

Visual Basic Copy Code

```
Public Property TemplatePath As String
```

Property Value

A string specifying the template file location.

See Also

VisioExporter Class

Exports diagrams to Visio 2003 XML Drawing files (*.vdx).

Syntax

C# Copy Code

```
public class VisioExporter
```

Visual Basic Copy Code

```
Public Class VisioExporter
```

Remarks

Add a reference to the *MindFusion.Diagramming.Export.Visio.dll* to your project in order to use this component.

The component allows exporting Visio 2003 XML Drawing files from MvcDiagram. The MvcDiagram ShapeNode objects are mapped to their closest Visio counterparts and TableNode objects are exported as groups of Visio shapes. MvcDiagram DiagramLink instances are exported as Visio connectors of a similar style. In addition, most of the attributes of diagram items such as text, text formatting, fill and line colors and styles are preserved.

To function properly, the exporter object requires the *VisioExport.vxt* file to be present in the application's directory. This file contains an XML template data used by the component to generate Visio documents.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Export.VisioExporter

See Also

VisioExporter Members
MindFusion.Diagramming.Export Namespace

VisioExporter Members

Exports diagrams to Visio 2003 XML Drawing files (*.vdx).

The following tables list the members exposed by the VisioExporter type.

Public Constructors

	Name	Description
	VisioExporter	Initializes a new instance of the **VisioExporter** class.

Public Properties

	Name	Description
	CreateVisioGroups	Gets or sets a value indicating whether Visio groups are created from the MvcDiagram groups.
	DynamicsOff	Gets or sets whether Visio connector rerouting should be disabled.
	ExportImagesAsGroups	Specifies whether a shape node that contains an image should be exported as a Visio group of a shape, picture and text objects.
	ExportInvisibleItems	Gets or sets a value indicating whether to export invisible items.

	ExportLanes	Gets or sets a value specifying whether the LaneGrid should be exported to Visio.
	ExportTablesAsGroups	Gets or sets a value indicating whether MvcDiagram TableNode objects should be exported as groups of Visio rectangular shapes.
	LaneGridTitle	Gets or sets the title of the functional band container created when exporting the diagram's swimlane grid.
	TemplatePath	Gets or sets the full path to the *VisioExport.vxt* template file.

Public Methods

	Name	Description
	ApproxColors	Approximate two color of gradient to get corresponded ACI.
	Export	Overloaded.

See Also

VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter Constructor

Initializes a new instance of the VisioExporter class.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public VisioExporter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter Methods

VisioExporter.ApproxColors Method

Approximate two color of gradient to get corresponded ACI.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public static Color ApproxColors (
    Color cr1,
    Color cr2
)
```

Visual Basic Copy Code

```
Public Shared Function ApproxColors( _
    cr1 As Color, _
    cr2 As Color _
) As Color
```

Parameters

cr1
Color 1 of gradient

cr2
Color 2 of gradient

Return Value

See Also

VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter.Export Method

Overload List

Name	Description
VisioExporter.Export (Diagram, Stream)	Creates a Visio XML Drawing file representing the specified MindFusion.Diagramming diagram.
VisioExporter.Export (Diagram, String)	Creates a Visio XML Drawing file representing the specified MvcDiagram diagram.
VisioExporter.Export (DiagramDocument, Stream)	Creates a Visio XML Drawing file representing the specified DiagramDocument object.
VisioExporter.Export (DiagramDocument, String)	Creates a Visio XML Drawing file representing the specified DiagramDocument object.

See Also

VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter.Export Method (Diagram, Stream)

Creates a Visio XML Drawing file representing the specified MindFusion.Diagramming diagram.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public void Export (
    Diagram diagram,
    Stream outStream
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagram As Diagram, _
    outStream As Stream _
)
```

Parameters

diagram

A reference to the diagram to be exported.

outStream

The stream to write to.

See Also

Export Method Overload List
VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter.Export Method (Diagram, String)

Creates a Visio XML Drawing file representing the specified MvcDiagram diagram.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public void Export (
    Diagram diagram,
    string vdxPathName
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagram As Diagram, _
    vdxPathName As String _
)
```

Parameters

diagram

A reference to the diagram to be exported.

vdxPathName

The full path to the exported Visio .vdx file.

Remarks

The exported document contains a single page, as big as the MvcDiagram scrollable diagram area. The Visio page dimension and items size measure unit is set to the diagram's MeasureUnit.

To run properly, this method requires the *VisioExport.vxt* file to be present in the application's directory. That file contains an XML template data used by the component to construct Visio documents.

See Also

Export Method Overload List
VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter.Export Method (DiagramDocument, Stream)

Creates a Visio XML Drawing file representing the specified DiagramDocument object.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public void Export (
    DiagramDocument diagramDoc,
    Stream outStream
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagramDoc As DiagramDocument, _
    outStream As Stream _
)
```

Parameters

diagramDoc
A reference to the document to be exported.

outStream
The stream to write to.

Remarks

Each DiagramPage in the document is exported to an individual Visio page. The title of the Visio page is set to DiagramPage.Title.

See Also

Export Method Overload List
VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter.Export Method (DiagramDocument, String)

Creates a Visio XML Drawing file representing the specified DiagramDocument object.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public void Export (
    DiagramDocument diagramDoc,
    string vdxPathName
)
```

Visual Basic Copy Code

```
Public Sub Export( _
    diagramDoc As DiagramDocument, _
    vdxPathName As String _
)
```

Parameters

diagramDoc
A reference to the document to be exported.

vdxPathName
Full path to the exported Visio .vdx file.

Remarks

Each DiagramPage in the document is exported to an individual Visio page. The title of the Visio page is set to DiagramPage.Title.

See Also

Export Method Overload List
VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter Properties

VisioExporter.CreateVisioGroups Property

Gets or sets a value indicating whether Visio groups are created from the MvcDiagram groups.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public bool CreateVisioGroups { get; set; }
```

Visual Basic Copy Code

```
Public Property CreateVisioGroups As Boolean
```

Property Value

true to create Visio groups when possible; otherwise, **false**. The default is **false**.

Remarks

It is not always possible to create a Visio group corresponding to a MvcDiagram one. If the structure of your diagram seems incorrect when imported in Visio, disable this property.

See Also

VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace
ExportTablesAsGroups Property

VisioExporter.DynamicsOff Property

Gets or sets whether Visio connector rerouting should be disabled.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public bool DynamicsOff { get; set; }
```

Visual Basic Copy Code

```
Public Property DynamicsOff As Boolean
```

Property Value

See Also

VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter.ExportImagesAsGroups Property

Specifies whether a shape node that contains an image should be exported as a Visio group of a shape, picture and text objects.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public bool ExportImagesAsGroups { get; set; }
```

Visual Basic Copy Code

```
Public Property ExportImagesAsGroups As Boolean
```

Property Value

true to export nodes with images as groups; otherwise, **false**. The default is **true**.

Remarks

Visio shapes cannot contain images inside them, and by default a MvcDiagram shape node that contains an image s exported as a Visio Picture object. Unfortunately, that leads to losing the node shape and the image alignment. Setting this property to **true** lets you export a shape node with an image as a group of several Visio objects that will preserve the appearance of the original MvcDiagram node as closely as possible.

See Also

VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace
ExportTablesAsGroups Property

VisioExporter.ExportInvisibleItems Property

Gets or sets a value indicating whether to export invisible items.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public bool ExportInvisibleItems { get; set; }
```

Visual Basic Copy Code

```
Public Property ExportInvisibleItems As Boolean
```

Property Value

true to export invisible items; otherwise, **false**.

See Also

VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter.ExportLanes Property

Gets or sets a value specifying whether the LaneGrid should be exported to Visio.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public ExportLanes ExportLanes { get; set; }
```

Visual Basic Copy Code

```
Public Property ExportLanes As ExportLanes
```

Property Value

A member of the ExportLanes enumeration. The default value is **None**.

Remarks

If export of swimlanes is enabled, they are exported as Visio *Functional Band* shapes.

See Also

VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter.ExportTablesAsGroups Property

Gets or sets a value indicating whether MvcDiagram TableNode objects should be exported as groups of Visio rectangular shapes.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public bool ExportTablesAsGroups { get; set; }
```

Visual Basic Copy Code

```
Public Property ExportTablesAsGroups As Boolean
```

Property Value

true to export tables as groups; otherwise, **false**. The default is **false**.

Remarks

Visio table entities cannot display cells in multiple columns. If your MvcDiagram diagrams contains tables having more than two columns, consider enabling this property. Then the VisioExporter would create a rectangular shape for each cell, and group the cells of each table into a Visio compound shape corresponding to the table.

See Also

VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace
ExportImagesAsGroups Property

VisioExporter.LaneGridTitle Property

Gets or sets the title of the functional band container created when exporting the diagram's swimlane grid.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public string LaneGridTitle { get; set; }
```

Visual Basic Copy Code

```
Public Property LaneGridTitle As String
```

Property Value

A string specifying the title of the Functional Band container shape. The default value is empty string.

Remarks

When ExportLanes is enabled, VisioExporter create a *Functional Band* shape for each swimlane and a container shape for the whole grid. This property specifies the container's title.

See Also

VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace

VisioExporter.TemplatePath Property

Gets or sets the full path to the *VisioExport.vxt* template file.

Namespace: MindFusion.Diagramming.Export
Assembly: MindFusion.Diagramming.Export.Visio

Syntax

C# Copy Code

```
public string TemplatePath { get; set; }
```

Visual Basic Copy Code

```
Public Property TemplatePath As String
```

Property Value

A string specifying the location of the template file. The default is an empty string.

Remarks

If you do not specify the path to the template file, VisioExporter looks for the file in the same folder where the *MindFusion.Diagramming.Export.Visio.dll* assembly is, in the folder of the application executable file, and in the current folder. However that might not work under restricted security permissions, for example when running in a browser or in an IIS process.

See Also

VisioExporter Members
VisioExporter Class
MindFusion.Diagramming.Export Namespace

MindFusion.Diagramming.Import Namespace

ConnectorImportedEventArgs Class

Contains the arguments passed to handlers of the ConnectorImported event.

Syntax

C# Copy Code

```
public class ConnectorImportedEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class ConnectorImportedEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.Import.ConnectorImportedEventArgs

See Also

ConnectorImportedEventArgs Members
MindFusion.Diagramming.Import Namespace

ConnectorImportedEventArgs Members

Contains the arguments passed to handlers of the ConnectorImported event.

The following tables list the members exposed by the ConnectorImportedEventArgs type.

Public Properties

	Name	Description
	ConnectorXml	Gets the XML DOM element corresponding to the Visio document's XML code that describes the imported connector.
	Link	Gets a reference to the imported diagram link.
	ShapeName	Gets the Visio name of the connector shape.

See Also

ConnectorImportedEventArgs Class
MindFusion.Diagramming.Import Namespace

ConnectorImportedEventArgs Properties

ConnectorImportedEventArgs.ConnectorXml Property

Gets the XML DOM element corresponding to the Visio document's XML code that describes the imported connector.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public XmlNode ConnectorXml { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ConnectorXml As XmlNode
```

Property Value

An **XmlNode** object corresponding to the section of the VDX document which describes the imported connector.

See Also

ConnectorImportedEventArgs Members
ConnectorImportedEventArgs Class
MindFusion.Diagramming.Import Namespace

ConnectorImportedEventArgs.Link Property

Gets a reference to the imported diagram link.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public DiagramLink Link { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Link As DiagramLink
```

Property Value

An instance of the DiagramLink class.

See Also

ConnectorImportedEventArgs Members
ConnectorImportedEventArgs Class
MindFusion.Diagramming.Import Namespace

ConnectorImportedEventArgs.ShapeName Property

Gets the Visio name of the connector shape.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public string ShapeName { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ShapeName As String
```

Property Value

A string specifying the Visio name of the connector.

See Also

ConnectorImportedEventArgs Members
ConnectorImportedEventArgs Class
MindFusion.Diagramming.Import Namespace

DrawImporter Class

DrawImporter lets you import an OpenOffice Draw document into a Diagram or DiagramDocument object.

Syntax

C# Copy Code

```
public class DrawImporter
```

Visual Basic Copy Code

```
Public Class DrawImporter
```

Remarks

The **DrawImporter** class can import .odg files, created by the OpenOffice Draw vector graphics editor. In order to use it, you must add a reference to the *MindFusion.Diagramming.Import.Draw.dll* assembly.

Various overloads of the Import method can be used to import the OpenOffice drawing into a DiagramDocument whose pages correspond to the Draw pages, or into a single Diagram whose content is merged from all imported pages. When merging multiple pages, their positions in the diagram depend on the PagesPerRow value.

DrawImporter supports shapes from the *General* and *Flowchart* shape palettes in Draw. For each node shape from the .odg file, the importer creates a ShapeNode object whose Shape, Text, Brush and Pen properties are set to values corresponding to the imported OpenOffice shape. For each connector shape, the importer creates a DiagramLink object whose Origin and Destination are set to the corresponding imported nodes, and whose ControlPoints reflect the geometry of the connector.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Import.DrawImporter

See Also

DrawImporter Members
MindFusion.Diagramming.Import Namespace

DrawImporter Members

DrawImporter lets you import an OpenOffice Draw document into a Diagram or DiagramDocument object.

The following tables list the members exposed by the DrawImporter type.

Public Constructors

	Name	Description
	DrawImporter	Initializes a new instance of the **DrawImporter** class.

Public Properties

	Name	Description
	PagesPerRow	Gets or sets how many Draw pages to import on a row before offsetting the import location vertically.

Public Methods

	Name	Description
	Import	Overloaded.

See Also

DrawImporter Class
MindFusion.Diagramming.Import Namespace

DrawImporter Constructor

Initializes a new instance of the DrawImporter class.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Draw

Syntax

C# Copy Code

```
public DrawImporter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DrawImporter Class
MindFusion.Diagramming.Import Namespace

DrawImporter Methods

DrawImporter.Import Method

Overload List

Name	Description
DrawImporter.Import (String)	Imports the specified OpenOffice Draw document into a new DiagramDocument, where each Draw page is imported into a separate DiagramPage of the **DiagramDocument**.
DrawImporter.Import (String, Diagram)	Imports the specified OpenOffice Draw document into the specified Diagram.
DrawImporter.Import (String, DiagramDocument)	Imports the specified OpenOffice Draw document into an existing DiagramDocument, where each Draw page is imported into a separate DiagramPage of the **DiagramDocument**.

See Also

DrawImporter Members
DrawImporter Class
MindFusion.Diagramming.Import Namespace

DrawImporter.Import Method (String)

Imports the specified OpenOffice Draw document into a new DiagramDocument, where each Draw page is imported into a separate DiagramPage of the **DiagramDocument**.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Draw

Syntax

C# Copy Code

```
public DiagramDocument Import (
    string filePath
)
```

Visual Basic Copy Code

```
Public Function Import( _
    filePath As String _
) As DiagramDocument
```

Parameters

filePath

A string specifying the path to the .odg file.

Return Value

A DiagramDocument containing the imported OpenOffice drawing.

See Also

Import Method Overload List
DrawImporter Members
DrawImporter Class
MindFusion.Diagramming.Import Namespace

DrawImporter.Import Method (String, Diagram)

Imports the specified OpenOffice Draw document into the specified Diagram.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Draw

Syntax

C# Copy Code

```
public void Import (
    string filePath,
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public Sub Import( _
    filePath As String, _
    diagram As Diagram _
)
```

Parameters

filePath

A string specifying the path to the .odg file.

diagram

A Diagram into which to import the OpenOffice drawing.

Remarks

Objects from each subsequent Draw page are imported in the same Diagram, at an offset depending on the page size and the value of the PagesPerRow property.

See Also

Import Method Overload List
DrawImporter Members

DrawImporter.Import Method (String, DiagramDocument)

Imports the specified OpenOffice Draw document into an existing DiagramDocument, where each Draw page is imported into a separate DiagramPage of the **DiagramDocument**.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Draw

Syntax

C# Copy Code

```
public void Import (
    string filePath,
    DiagramDocument document
)
```

Visual Basic Copy Code

```
Public Sub Import( _
    filePath As String, _
    document As DiagramDocument _
)
```

Parameters

filePath

A string specifying the path to the .odg file.

document

A DiagramDocument into which to import the OpenOffice drawing.

See Also

Import Method Overload List
DrawImporter Members
DrawImporter Class
MindFusion.Diagramming.Import Namespace

DrawImporter Properties

DrawImporter.PagesPerRow Property

Gets or sets how many Draw pages to import on a row before offsetting the import location vertically.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Draw

Syntax

C# Copy Code

```
public int PagesPerRow { get; set; }
```

Visual Basic Copy Code

```
Public Property PagesPerRow As Integer
```

Property Value

An integer value specifying how many pages to fit horizontally. The default value is 1.

Remarks

This property is considered by the Import overload that imports all pages into a single diagram.

See Also

DrawImporter Members
DrawImporter Class
MindFusion.Diagramming.Import Namespace

GroupImportedEventArgs Class

Contains the arguments passed to handlers of the GroupImported event.

Syntax

C# Copy Code

```
public class GroupImportedEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class GroupImportedEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.Import.GroupImportedEventArgs

See Also

GroupImportedEventArgs Members
MindFusion.Diagramming.Import Namespace

GroupImportedEventArgs Members

Contains the arguments passed to handlers of the GroupImported event.

The following tables list the members exposed by the GroupImportedEventArgs type.

Public Properties

	Name	Description
	Group	Gets a reference to the imported group.

See Also

GroupImportedEventArgs Class
MindFusion.Diagramming.Import Namespace

GroupImportedEventArgs Properties

GroupImportedEventArgs.Group Property

Gets a reference to the imported group.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public Group Group { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Group As Group
```

Property Value

An instance of the Group class.

See Also

GroupImportedEventArgs Members
GroupImportedEventArgs Class
MindFusion.Diagramming.Import Namespace

ImportItemEventArgs Class

Contains the arguments passed to handlers of the ImportItem event.

Syntax

C# Copy Code

```
public class ImportItemEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class ImportItemEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.Import.ImportItemEventArgs

See Also

ImportItemEventArgs Members
MindFusion.Diagramming.Import Namespace

ImportItemEventArgs Members

Contains the arguments passed to handlers of the ImportItem event.

The following tables list the members exposed by the ImportItemEventArgs type.

Public Properties

	Name	Description
	Item	Gets or sets the item that should represent the Visio shape specified by ShapeXml in the imported diagram.
	MasterName	Gets the name associated with shape's master.
	MasterXml	Gets the XML DOM element corresponding to the Visio document's XML code that describes the shape master.
	NamespaceManager	Gets an **XmlNamespaceManager** instance with preloaded Visio namespaces that can be used when querying the XML elements provided

		by this **ImportItemEventArgs** instance.
	ShapeXml	Gets the XML DOM element corresponding to the Visio document's XML code that describes the imported shape.

See Also

ImportItemEventArgs Class
MindFusion.Diagramming.Import Namespace

ImportItemEventArgs Properties

ImportItemEventArgs.Item Property

Gets or sets the item that should represent the Visio shape specified by ShapeXml in the imported diagram.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public DiagramItem Item { get; set; }
```

Visual Basic Copy Code

```
Public Property Item As DiagramItem
```

Property Value

A DiagramItem instance representing the Visio shape currently being imported.

See Also

ImportItemEventArgs Members
ImportItemEventArgs Class
MindFusion.Diagramming.Import Namespace

ImportItemEventArgs.MasterName Property

Gets the name associated with shape's master.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public string MasterName { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MasterName As String
```

Property Value

A string specifying the name of the shape master.

See Also

ImportItemEventArgs Members
ImportItemEventArgs Class
MindFusion.Diagramming.Import Namespace

ImportItemEventArgs.MasterXml Property

Gets the XML DOM element corresponding to the Visio document's XML code that describes the shape master.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public XmlNode MasterXml { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property MasterXml As XmlNode
```

Property Value

A **System.Xml.XmlNode** instance.

See Also

ImportItemEventArgs Members
ImportItemEventArgs Class
MindFusion.Diagramming.Import Namespace

ImportItemEventArgs.NamespaceManager Property

Gets an **XmlNamespaceManager** instance with preloaded Visio namespaces that can be used when querying the XML elements provided by this ImportItemEventArgs instance.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public XmlNamespaceManager NamespaceManager { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NamespaceManager As XmlNamespaceManager
```

Property Value

A **System.Xml.XmlNamespaceManager** instance.

See Also

ImportItemEventArgs Members
ImportItemEventArgs Class
MindFusion.Diagramming.Import Namespace

ImportItemEventArgs.ShapeXml Property

Gets the XML DOM element corresponding to the Visio document's XML code that describes the imported shape.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public XmlNode ShapeXml { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ShapeXml As XmlNode
```

Property Value

A **System.Xml.XmlNode** instance.

See Also

ImportItemEventArgs Members
ImportItemEventArgs Class
MindFusion.Diagramming.Import Namespace

ShapeImportedEventArgs Class

Contains the arguments passed to handlers of the ShapeImported event.

Syntax

C# Copy Code

```
public class ShapeImportedEventArgs : EventArgs
```

Visual Basic Copy Code

```
Public Class ShapeImportedEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

System.Object
  System.EventArgs
    MindFusion.Diagramming.Import.ShapeImportedEventArgs

See Also

ShapeImportedEventArgs Members
MindFusion.Diagramming.Import Namespace

ShapeImportedEventArgs Members

Contains the arguments passed to handlers of the ShapeImported event.

The following tables list the members exposed by the ShapeImportedEventArgs type.

Public Properties

	Name	Description
	Shape	Gets a reference to the imported diagram shape.
	ShapeName	Gets the Visio name of the node shape.
	ShapeXml	Gets the XML DOM element corresponding to the Visio document's XML code that describes the imported shape.

See Also

ShapeImportedEventArgs Properties

ShapeImportedEventArgs.Shape Property

Gets a reference to the imported diagram shape.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public ShapeNode Shape { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Shape As ShapeNode
```

Property Value

An instance of the ShapeNode class.

See Also

ShapeImportedEventArgs Members
ShapeImportedEventArgs Class
MindFusion.Diagramming.Import Namespace

ShapeImportedEventArgs.ShapeName Property

Gets the Visio name of the node shape.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public string ShapeName { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ShapeName As String
```

Property Value

A string specifying the Visio name of the shape.

See Also

ShapeImportedEventArgs Members
ShapeImportedEventArgs Class
MindFusion.Diagramming.Import Namespace

ShapeImportedEventArgs.ShapeXml Property

Gets the XML DOM element corresponding to the Visio document's XML code that describes the imported shape.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public XmlNode ShapeXml { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ShapeXml As XmlNode
```

Property Value

An **XmlNode** object corresponding to the section of the VDX document which describes the imported shape.

Example

The following sample shows how to read a custom property value defined in Visio by handling the ShapeImported event.

C# Copy Code

```
private void ShapeImported(object sender, ShapeImportedEventArgs e)
{
    XmlNode vdxCode = e.ShapeXml;
    try
    {
        XmlNamespaceManager ns = new XmlNamespaceManager(e.ShapeXml.OwnerDocument.NameTable);
        ns.AddNamespace("vdx", "http://schemas.microsoft.com/visio/2003/core");

        XmlNode myPropVal =
vdxCode.SelectSingleNode("vdx:Prop[vdx:Label='MyProperty']/vdx:Value", ns);
        if (myPropVal != null)
            e.Shape.Tag = myPropVal.InnerText;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

See Also

ShapeImportedEventArgs Members
ShapeImportedEventArgs Class
MindFusion.Diagramming.Import Namespace

SvgImporter Class

SvgImporter lets you import the contents of an SVG file into a Diagram object.

Syntax

C# Copy Code

```
public class SvgImporter
```

Visual Basic Copy Code

```
Public Class SvgImporter
```

Remarks

Add a reference to the *MindFusion.Diagramming.Import.Svg.dll* to your project in order to use this component.

The component allows importing Scalable Vector Graphics files (*.svg) into MindFusion.Diagramming. Each top-level SVG group element is imported as a separate SvgNode, whose Content is set to the group's child elements. If the SVG file was exported from

Visio or OpenOffice Draw, **SvgImporter** will create DiagramLink objects for groups it recognizes as connectors in order to recreate the original diagram model, though in this case it will not preserve fully the original appearance of connectors in SVG.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Import.SvgImporter

See Also

SvgImporter Members
MindFusion.Diagramming.Import Namespace

SvgImporter Members

SvgImporter lets you import the contents of an SVG file into a Diagram object.

The following tables list the members exposed by the SvgImporter type.

Public Constructors

	Name	Description
	SvgImporter	Initializes a new instance of the SvgImporter class.

Public Properties

	Name	Description
	TopLevelGroupIsPage	Gets or sets whether top-level SVG groups should be treated as page content.

Public Methods

	Name	Description
	Import	Imports the specified SVG file into the specified Diagram.

See Also

SvgImporter Class
MindFusion.Diagramming.Import Namespace

SvgImporter Constructor

Initializes a new instance of the SvgImporter class.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Svg

Syntax

C# Copy Code

```
public SvgImporter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SvgImporter Class
MindFusion.Diagramming.Import Namespace

SvgImporter Methods

SvgImporter.Import Method

Imports the specified SVG file into the specified Diagram.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Svg

Syntax

C# Copy Code

```
public void Import (
    string filePath,
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public Sub Import( _
    filePath As String, _
    diagram As Diagram _
)
```

Parameters

filePath
A string specifying the path to the .svg file.

diagram
A Diagram into which to import the SVG drawing.

See Also

SvgImporter Members
SvgImporter Class
MindFusion.Diagramming.Import Namespace

SvgImporter Properties

SvgImporter.TopLevelGroupIsPage Property

Gets or sets whether top-level SVG groups should be treated as page content.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Svg

Syntax

C# Copy Code

```
public bool TopLevelGroupIsPage { get; set; }
```

Visual Basic Copy Code

```
Public Property TopLevelGroupIsPage As Boolean
```

Property Value

See Also

Visio2013Importer Class

Imports Visio 2013 XML Drawing files (*.vsdx) into Diagram or DiagramDocument objects.

Syntax

C# Copy Code

```
public class Visio2013Importer
```

Visual Basic Copy Code

```
Public Class Visio2013Importer
```

Remarks

The **Visio2013Importer** class can import .vsdx files, created by version 2013 of Microsoft Visio. In order to use it, you must add a reference to the *MindFusion.Diagramming.Import.Visio.dll* assembly.

Various overloads of the Import method can be used to import the Visio drawing into a DiagramDocument whose pages correspond to the Visio pages, or into a single Diagram whose content is merged from all imported pages. When merging multiple pages, their positions in the diagram depend on the PagesPerRow value.

Visio2013Importer supports Visio's basic flowchart and data flow stencils. For each node shape from the .vsdx file, the importer creates a ShapeNode object whose Shape, Text, Brush and Pen properties are set to values corresponding to the imported Visio shape. For each connector shape, the importer creates a DiagramLink object whose Origin and Destination are set to the corresponding imported nodes, and whose ControlPoints reflect the geometry of the connector.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Import.Visio2013Importer

See Also

Visio2013Importer Members
MindFusion.Diagramming.Import Namespace

Visio2013Importer Members

Imports Visio 2013 XML Drawing files (*.vsdx) into Diagram or DiagramDocument objects.

The following tables list the members exposed by the Visio2013Importer type.

Public Constructors

	Name	Description
	Visio2013Importer	Initializes a new instance of the Visio2013Importer class.

Public Properties

	Name	Description
	PagesPerRow	Gets or sets how many Visio pages to import on a row before offsetting the import location vertically.

Public Methods

	Name	Description
	Import	Overloaded.

Protected Methods

	Name	Description
	OnImportItem	Raises the ImportItem event.

Public Events

	Name	Description
	ImportItem	Raised for each shape's XML element in the imported VSDX drawing to let you create a custom diagram item representing the Visio shape.

See Also

Visio2013Importer Class
MindFusion.Diagramming.Import Namespace

Visio2013Importer Constructor

Initializes a new instance of the Visio2013Importer class.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public Visio2013Importer ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

Visio2013Importer Class
MindFusion.Diagramming.Import Namespace

Visio2013Importer Methods

Visio2013Importer.Import Method

Overload List

Name	Description
Visio2013Importer.Import (String)	Imports the specified Visio Graphics Service VSDX file into a new DiagramDocument, where each Visio page is imported into a separate DiagramPage of the DiagramDocument.
Visio2013Importer.Import (String, Diagram)	Imports the specified Visio Graphics Service VSDX file into the specified Diagram.
Visio2013Importer.Import (String, DiagramDocument)	Imports the specified Visio Graphics Service VSDX file into an existing DiagramDocument, where each Visio page is imported into a separate DiagramPage of the DiagramDocument.

See Also

Visio2013Importer.Import Method (String)

Imports the specified Visio Graphics Service VSDX file into a new DiagramDocument, where each Visio page is imported into a separate DiagramPage of the DiagramDocument.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public DiagramDocument Import (
    string filePath
)
```

Visual Basic Copy Code

```
Public Function Import( _
    filePath As String _
) As DiagramDocument
```

Parameters

filePath
A string specifying the path to the .vsdx file.

Return Value

A DiagramDocument containing the imported Visio drawing.

See Also

Import Method Overload List
Visio2013Importer Members
Visio2013Importer Class
MindFusion.Diagramming.Import Namespace

Visio2013Importer.Import Method (String, Diagram)

Imports the specified Visio Graphics Service VSDX file into the specified Diagram.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public void Import (
    string filePath,
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public Sub Import( _
    filePath As String, _
    diagram As Diagram _
)
```

Parameters

filePath
A string specifying the path to the .vsdx file.

diagram
A Diagram into which to import the Visio drawing.

See Also

Import Method Overload List
Visio2013Importer Members
Visio2013Importer Class
MindFusion.Diagramming.Import Namespace

Visio2013Importer.Import Method (String, DiagramDocument)

Imports the specified Visio Graphics Service VSDX file into an existing DiagramDocument, where each Visio page is imported into a separate DiagramPage of the DiagramDocument.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public void Import (
    string filePath,
    DiagramDocument document
)
```

Visual Basic Copy Code

```
Public Sub Import( _
    filePath As String, _
    document As DiagramDocument _
)
```

Parameters

filePath
A string specifying the path to the .vsdx file.

document
A DiagramDocument into which to import the Visio drawing.

See Also

Import Method Overload List
Visio2013Importer Members
Visio2013Importer Class
MindFusion.Diagramming.Import Namespace

Visio2013Importer.OnImportItem Method

Raises the ImportItem event.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
protected virtual void OnImportItem (
    Object sender,
    ImportItemEventArgs args
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnImportItem( _
    sender As Object, _
    args As ImportItemEventArgs _
)
```

Parameters

sender

args

See Also

Visio2013Importer Members
Visio2013Importer Class
MindFusion.Diagramming.Import Namespace

Visio2013Importer Properties

Visio2013Importer.PagesPerRow Property

Gets or sets how many Visio pages to import on a row before offsetting the import location vertically.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public int PagesPerRow { get; set; }
```

Visual Basic Copy Code

```
Public Property PagesPerRow As Integer
```

Property Value

See Also

Visio2013Importer Members
Visio2013Importer Class
MindFusion.Diagramming.Import Namespace

Visio2013Importer Events

Visio2013Importer.ImportItem Event

Raised for each shape's XML element in the imported VSDX drawing to let you create a custom diagram item representing the Visio shape.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public event EventHandler<ImportItemEventArgs> ImportItem
```

Visual Basic Copy Code

```
Public Event ImportItem As EventHandler(Of ImportItemEventArgs)
```

See Also

Visio2013Importer Members
Visio2013Importer Class
MindFusion.Diagramming.Import Namespace

VisioImporter Class

Imports Visio 2003 XML Drawing files (*.vdx) into MvcDiagram.

Syntax

C# Copy Code

```
public class VisioImporter
```

Visual Basic Copy Code

```
Public Class VisioImporter
```

Remarks

Add a reference to the *MindFusion.Diagramming.Import.Visio.dll* to your project in order to use this component.

The component allows importing Visio 2003 XML Drawing files (*.vdx) into MvcDiagram. The standard Visio flowchart shapes, as well as most of the UML and Database entity shapes are imported to their closest representation in MvcDiagram. The Visio connectors are imported as MvcDiagram DiagramLink objects, preserving the relationship between the connected nodes. In addition, most of the attributes of Visio objects such as text, text formatting, fill and line colors and styles are preserved. The component raises an event for each imported node to let you set the node shape if it is not recognized by MvcDiagram or if you need to set a different shape.

To determine how many pages contains the Visio document and import only a single page, call the GetPageCount and ImportPage methods. All pages of a Visio diagram can be imported at once by calling the ImportAllPages method, which lays out the pages as set in PagesPerRow. More properties and events provided by the class are listed below.

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Import.VisioImporter

See Also

VisioImporter Members
MindFusion.Diagramming.Import Namespace

VisioImporter Members

Imports Visio 2003 XML Drawing files (*.vdx) into MvcDiagram.

The following tables list the members exposed by the VisioImporter type.

Public Constructors

	Name	Description
	VisioImporter	Initializes a new instance of the **VisioImporter** class.

Public Properties

	Name	Description
	ApplyVisioUnit	Gets or sets a value indicating whether the diagram's MeasureUnit should be set to the one read from the imported Visio file.

	ImportEntitiesAsTables	Gets or sets a value indicating whether Visio UML and DB entities are imported as MvcDiagram tables.
	PagesPerRow	Gets or sets how many Visio pages to import on a row before offsetting the import location vertically.
	SuppressUnmanagedCode	Suppress usage of unmanaged code.

Public Methods

	Name	Description
	GetPageCount	Determines how many pages are there in a Visio diagram.
	Import	Imports all pages from the specified Visio document into a single DiagramDocument object.
	ImportAllPages	Imports all pages from the specified Visio document.
	ImportPage	Imports a single page from the specified Visio document.

Protected Methods

	Name	Description
	OnConnectorImported	Raises the ConnectorImported event.
	OnGroupImported	Raises the GroupImported event.
	OnShapeImported	Raises the ShapeImported event.

Public Events

	Name	Description
	ConnectorImported	Raised when a Visio connector is imported as a MvcDiagram link.
	GroupImported	Raised when a Visio group is imported as a MvcDiagram group.
	ShapeImported	Raised when a Visio shape is imported into MvcDiagram as a ShapeNode.

See Also

VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter Constructor

Initializes a new instance of the VisioImporter class.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public VisioImporter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

VisioImporter Methods

VisioImporter.GetPageCount Method

Determines how many pages are there in a Visio diagram.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public long GetPageCount (
    string vdxPathName
)
```

Visual Basic Copy Code

```
Public Function GetPageCount( _
    vdxPathName As String _
) As Long
```

Parameters

vdxPathName
A full path to the Visio VDX file.

Return Value

An integer value specifying the number of pages in the Visio diagram.

Remarks

Use this method to determine the valid range of pages for ImportPage.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter.Import Method

Imports all pages from the specified Visio document into a single DiagramDocument object.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public DiagramDocument Import (
    string vdxPathName
)
```

Visual Basic Copy Code

```
Public Function Import( _
    vdxPathName As String _
) As DiagramDocument
```

Parameters

vdxPathName
A full path to the Visio VDX file.

Return Value

The newly crated DiagramDocument object.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter.ImportAllPages Method

Imports all pages from the specified Visio document.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public void ImportAllPages (
    string vdxPathName,
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public Sub ImportAllPages ( _
    vdxPathName As String, _
    diagram As Diagram _
)
```

Parameters

vdxPathName
A full path to the Visio VDX file.

diagram
Reference to the diagram where the Visio drawing is imported.

Remarks

The pages are imported starting from the top-left point of the MvcDiagram diagram area. The current import location is moved one page-width to the right with each new page imported, until PagesPerRow pages are imported on the same y-coordinate. When that happens, the current location moves to the left-most x-coordinate and one page-height down. If there are more pages to import, the process continues as above.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter.ImportPage Method

Imports a single page from the specified Visio document.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public void ImportPage (
    string vdxPathName,
    Diagram diagram,
    int PageIndex
)
```

Visual Basic Copy Code

```
Public Sub ImportPage ( _
    vdxPathName As String, _
    diagram As Diagram, _
    PageIndex As Integer _
)
```

Parameters

vdxPathName
A full path to the Visio VDX file.

diagram
Reference to the Diagram document into which the Visio data is imported.

pageIndex
Specifies which page of the Visio document to import.

Remarks

The page index is zero based. Use GetPageCount to determine how many pages are there in the Visio diagram.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter.OnConnectorImported Method

Raises the ConnectorImported event.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
protected virtual void OnConnectorImported (
    ConnectorImportedEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnConnectorImported ( _
    e As ConnectorImportedEventArgs _
)
```

Parameters

e
An ConnectorImportedEventArgs that contains the event data.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter.OnGroupImported Method

Raises the GroupImported event.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
protected virtual void OnGroupImported (
    GroupImportedEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnGroupImported ( _
    e As GroupImportedEventArgs _
)
```

Parameters

e
An GroupImportedEventArgs that contains the event data.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter.OnShapeImported Method

Raises the ShapeImported event.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
protected virtual void OnShapeImported (
    ShapeImportedEventArgs e
)
```

Visual Basic Copy Code

```
Protected Overridable Sub OnShapeImported ( _
    e As ShapeImportedEventArgs _
)
```

Parameters

e
An ShapeImportedEventArgs that contains the event data.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter Properties

VisioImporter.ApplyVisioUnit Property

Gets or sets a value indicating whether the diagram's MeasureUnit should be set to the one read from the imported Visio file.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public bool ApplyVisioUnit { get; set; }
```

Visual Basic Copy Code

```
Public Property ApplyVisioUnit As Boolean
```

Property Value

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter.ImportEntitiesAsTables Property

Gets or sets a value indicating whether Visio UML and DB entities are imported as MvcDiagram tables.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public bool ImportEntitiesAsTables { get; set; }
```

Visual Basic Copy Code

```
Public Property ImportEntitiesAsTables As Boolean
```

Property Value

true to import Visio entity nodes as tables; otherwise, **false**. The default is **false**.

Remarks

If this property is disabled, Visio entity objects are imported as MvcDiagram groups of rectangular nodes; if it is enabled, entities are imported as tables.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter.PagesPerRow Property

Gets or sets how many Visio pages to import on a row before offsetting the import location vertically.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public long PagesPerRow { get; set; }
```

Visual Basic Copy Code

```
Public Property PagesPerRow As Long
```

Property Value

An integer value specifying how many pages to import in a row. The default is 1.

Remarks

MvcDiagram does not support stacked pages as Visio does, so the ImportAllPages method imports all pages from a Visio document in a grid that prevents items from one page to overlap with items from another page. This property specifies the width of the page grid.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter.SuppressUnmanagedCode Property

Suppress usage of unmanaged code.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public bool SuppressUnmanagedCode { get; set; }
```

Visual Basic Copy Code

```
Public Property SuppressUnmanagedCode As Boolean
```

Property Value

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter Events

VisioImporter.ConnectorImported Event

Raised when a Visio connector is imported as a MvcDiagram link.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C# Copy Code

```
public event EventHandler<ConnectorImportedEventArgs> ConnectorImported
```

Visual Basic	Copy Code
`Public Event ConnectorImported As EventHandler(Of ConnectorImportedEventArgs)`	

Event Data

ConnectorImported event handlers receive an argument of type ConnectorImportedEventArgs. The following **ConnectorImportedEventArgs** members provide information relevant to the event:

Member Name	Description
Link	Refers to the imported DiagramLink instance.
ShapeName	The name of the Visio connector shape.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter.GroupImported Event

Raised when a Visio group is imported as a MvcDiagram group.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C#	Copy Code
`public event EventHandler<GroupImportedEventArgs> GroupImported`	

Visual Basic	Copy Code
`Public Event GroupImported As EventHandler(Of GroupImportedEventArgs)`	

Event Data

GroupImported event handlers receive an argument of type GroupImportedEventArgs. The following **GroupImportedEventArgs** members provide information relevant to the event:

Member name	Description
Group	Refers to the imported Group instance.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

VisioImporter.ShapeImported Event

Raised when a Visio shape is imported into MvcDiagram as a ShapeNode.

Namespace: MindFusion.Diagramming.Import
Assembly: MindFusion.Diagramming.Import.Visio

Syntax

C#	Copy Code
`public event EventHandler<ShapeImportedEventArgs> ShapeImported`	

Visual Basic Copy Code

```
Public Event ShapeImported As EventHandler(Of ShapeImportedEventArgs)
```

Event Data

ShapeImported event handlers receive an argument of type ShapeImportedEventArgs. The following **ShapeImportedEventArgs** members provide information relevant to the event:

Member name	Description
Shape	Refers to the imported ShapeNode instance.
ShapeName	The name of the Visio node shape.

Remarks

The basic **Visio** flowchart shapes are mapped to the MvcDiagram predefined shape templates. If the importer does not recognize a specific Visio shape, the node created for it is set to a simple rectangle. You can handle this event in order to set the shape of a node whose Visio counterpart is not recognized, or whose default value set by the importer is not appropriate for your application.

See Also

VisioImporter Members
VisioImporter Class
MindFusion.Diagramming.Import Namespace

MindFusion.Diagramming.Layout Namespace

AnnealLayout Class

Implements a simulated annealing graph layout algorithm.

Syntax

C# Copy Code

```
public class AnnealLayout : Layout
```

Visual Basic Copy Code

```
Public Class AnnealLayout
    Inherits Layout
```

Remarks

Simulated Annealing is a general-purpose optimization method used to solve large-scale combinatorial problems by simulating the process of heating and cooling of metal to achieve freedom from defects. Finding a nice arrangement of a graph is a combinatorial problem that can be reduced to assigning costs to graph configurations and finding the minimum cost configuration. In that case, a cost is assigned to a graph configuration by evaluating different aesthetic criteria such as distance between nodes, length of links and the number of link crossings.

The **AnnealLayout** class implements a Simulated Annealing graph layout algorithm that can be used to arrange MvcDiagram diagrams. To arrange a diagram, create an **AnnealLayout** instance, set its properties and call the Arrange method.

The algorithm starts the simulation with the temperature set in InitialTemperature, and runs several cooling Stages. At each stage the algorithm evaluates several graph configurations for each node, as set in IterationsPerStage. At the end of each stage, the algorithm selects the configuration that has a minimum cost and decreases the temperature by multiplying it by TemperatureScale. The relative weight of an aesthetic criterion relative to the other criteria is set by the DistributionFactor, LinkLengthFactor, BoundaryFactor and CrossingLinksCost properties.

Anchoring specifies whether the layout algorithm must attach links to the anchor points of graphs nodes. The layout algorithm produces best results if it can place link ends at arbitrary positions over the nodes outlines, that is, when using the Ignore value. If your application logic requires links to be always attached to anchor points, use the other two values. In such cases **Reassign** would yield better results.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.AnnealLayout

See Also

AnnealLayout Members
MindFusion.Diagramming.Layout Namespace

AnnealLayout Members

Implements a simulated annealing graph layout algorithm.

The following tables list the members exposed by the AnnealLayout type.

Public Constructors

	Name	Description
	AnnealLayout	Initializes a new instance of the **AnnealLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	ApproximateLinkLength	Gets or sets a value indicating whether the layout should approximate the length of graph links by calculating the distance between the centers of their incident nodes.
	BoundaryFactor	Gets or sets a value indicating the importance of the distance between nodes and the layout area boundaries, relative to the other criteria considered by the algorithm.
	CrossingLinksCost	Gets or sets a value indicating the importance of the low number of link crossings, relative to the other criteria considered by the algorithm.
	DistributionFactor	Gets or sets a value indicating the importance of the node distribution, relative to the other criteria considered by the algorithm.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GatherStatistics	
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	InitialTemperature	Gets or sets the initial temperature of the simulated annealing process.
	IterationsPerStage	Gets or sets how many node shift iterations to perform at each stage of the algorithm.
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutArea	Gets or sets the size of the layout area.
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)

	LinkLengthFactor	Gets or sets a value indicating the importance of the short length of links, relative to the other criteria considered by the algorithm.
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	NodeLinkCrossingCost	Gets or sets a value indicating the importance of low number of overlaps between links and nodes relative to the other criteria considered by the algorithm.
	NodeLinkDistFactor	Gets or sets a value indicating the importance of the distance between nodes and links, relative to the other criteria considered by the algorithm.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	Randomize	Gets or sets a value indicating whether the nodes should be placed at random positions when the layout routine starts.
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	Root	Gets or sets a node indicating which subgraph of the diagram should be laid out.
	SplitGraph	Gets or sets a value indicating whether unconnected subgraphs should be laid out independently from each other.
	Stages	Gets or sets how many cool-down stages the algorithm should simulate.
	Statistics	
	TemperatureScale	Gets or sets how much the simulated temperature is decreased at each stage of the algorithm.
	WidthHeightRatio	Gets or sets what width/height ratio the layout area should have.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)
	BeginArrange	Initializes the layout object for a step-by-step execution.
	EndArrange	Completes the step-by-step execution started with a call to **BeginArrange**.

	Iterate	Runs one or more layout iterations.

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout Constructor

Initializes a new instance of the AnnealLayout class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public AnnealLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout Methods

AnnealLayout.BeginArrange Method

Initializes the layout object for a step-by-step execution.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool BeginArrange (
    Diagram diagram,
    DiagramItemCollection items
)
```

Visual Basic Copy Code

```
Public Function BeginArrange( _
    diagram As Diagram, _
    items As DiagramItemCollection _
) As Boolean
```

Parameters

diagram

The Diagram that will be arranged.

items

A collection of nodes and links from *diagram* that define the subgraph to arrange.

Return Value

true if the layout object has been initialized successfully; otherwise, **false**.

Remarks

You can use this method together with Iterate and EndArrange to display an animation of the layout process. To run all iterations of the process at once and display just the end result, call Arrange.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.EndArrange Method

Completes the step-by-step execution started with a call to BeginArrange.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void EndArrange ()
```

Visual Basic Copy Code

```
Public Sub EndArrange ()
```

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.Iterate Method

Runs one or more layout iterations.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void Iterate (
    int iterationFrom,
    int iterationTo
)
```

Visual Basic Copy Code

```
Public Sub Iterate( _
    iterationFrom As Integer, _
    iterationTo As Integer _
)
```

Parameters

iterationFrom

Specifies from which iteration to start.

iterationTo

Specifies at which iteration to stop.

Remarks

Use this method to create an animation of the layout process, for example, by calling it from a timer. Call BeginArrange to prepare the layout object for use of **Iterate**, and call EndArrange when done. Each call to **Iterate** should start from the iteration next to the one where the previous call has stopped.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout Properties

AnnealLayout.ApproximateLinkLength Property

Gets or sets a value indicating whether the layout should approximate the length of graph links by calculating the distance between the centers of their incident nodes.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool ApproximateLinkLength { get; set; }
```

Visual Basic Copy Code

```
Public Property ApproximateLinkLength As Boolean
```

Property Value

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.BoundaryFactor Property

Gets or sets a value indicating the importance of the distance between nodes and the layout area boundaries, relative to the other criteria considered by the algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double BoundaryFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property BoundaryFactor As Double
```

Property Value

A double multiplier applied to the node-to-boundary distance cost when evaluating the total cost of a graph configuration. The default is 3000.

Remarks

Assigning a larger value to this property will make the algorithm choose graph configurations in which nodes are placed farther from the layout area bounds, that is, nodes will tend to gather in the center of the layout area.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace
LinkLengthFactor Property
DistributionFactor Property

AnnealLayout.CrossingLinksCost Property

Gets or sets a value indicating the importance of the low number of link crossings, relative to the other criteria considered by the algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double CrossingLinksCost { get; set; }
```

Visual Basic Copy Code

```
Public Property CrossingLinksCost As Double
```

Property Value

A double value added to the total cost of a graph configuration for each pair of crossing links. The default is 120000.

Remarks

Assigning a larger value to this property will make the algorithm choose graph configurations with fewer link crossings.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace
LinkLengthFactor Property

AnnealLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.DistributionFactor Property

Gets or sets a value indicating the importance of the node distribution, relative to the other criteria considered by the algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double DistributionFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property DistributionFactor As Double
```

Property Value

A double multiplier applied to the node distribution cost when evaluating the total cost of a graph configuration. The default is 40000.

Remarks

Assigning a larger value to this property will make the algorithm choose graph configurations in which the nodes are placed farther one from another.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace
LinkLengthFactor Property
BoundaryFactor Property

AnnealLayout.GatherStatistics Property

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool GatherStatistics { get; set; }
```

Visual Basic Copy Code

```
Public Property GatherStatistics As Boolean
```

Property Value

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.InitialTemperature Property

Gets or sets the initial temperature of the simulated annealing process.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double InitialTemperature { get; set; }
```

Visual Basic Copy Code

```
Public Property InitialTemperature As Double
```

Property Value

A double value specifying the initial temperature of the simulation. The default is 40.

Remarks

A low initial temperature prevents nodes from moving far from their initial position. If you need the final arrangement to have resemblance to the initial formation of the graph elements, set a lower initial temperature and disable Randomize.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace
TemperatureScale Property

AnnealLayout.IterationsPerStage Property

Gets or sets how many node shift iterations to perform at each stage of the algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int IterationsPerStage { get; set; }
```

Visual Basic Copy Code

```
Public Property IterationsPerStage As Integer
```

Property Value

An integer value specifying the number of node shift iterations. The default is 50.

Remarks

IterationsPerStage specifies how many node configurations to evaluate at each stage of the temperature cool-down simulation. More iterations will yield better layouts but require more time to complete.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace
Progress Property
Stages Property

AnnealLayout.LayoutArea Property

Gets or sets the size of the layout area.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Rect LayoutArea { get; set; }
```

Visual Basic Copy Code

```
Public Property LayoutArea As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** representing the boundaries of the layout area. The default is **Empty**.

Remarks

If the layout area is not explicitly set, the algorithm will choose one suitable for the graph size. In that case the WidthHeightRatio property specifies how much larger the layout area width should be than its height.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.LinkLengthFactor Property

Gets or sets a value indicating the importance of the short length of links, relative to the other criteria considered by the algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double LinkLengthFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkLengthFactor As Double
```

Property Value

A double multiplier applied to the link length cost when evaluating the total cost of a graph configuration. The default is 0.25.

Remarks

Assigning a larger value to this property will make the algorithm choose graph configurations in which links are shorter.

See Also

AnnealLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Anneal layout".

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.NodeLinkCrossingCost Property

Gets or sets a value indicating the importance of low number of overlaps between links and nodes relative to the other criteria considered by the algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double NodeLinkCrossingCost { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeLinkCrossingCost As Double
```

Property Value

A double value added to the total cost of a graph configuration when a link crosses a node. The default is 15000.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.NodeLinkDistFactor Property

Gets or sets a value indicating the importance of the distance between nodes and links, relative to the other criteria considered by the algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double NodeLinkDistFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeLinkDistFactor As Double
```

Property Value

A double value specifying the distance factor.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.Randomize Property

Gets or sets a value indicating whether the nodes should be placed at random positions when the layout routine starts.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool Randomize { get; set; }
```

Visual Basic Copy Code

```
Public Property Randomize As Boolean
```

Property Value

true to start the simulation with random node positions; **false** to start it with the original node positions. The default is **true**.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.Root Property

Gets or sets a node indicating which subgraph of the diagram should be laid out.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public DiagramNode Root { get; set; }
```

Visual Basic Copy Code

```
Public Property Root As DiagramNode
```

Property Value

A DiagramNode whose containing graph should be arranged. The default is **null** (**Nothing** in Visual Basic).

Remarks

If a root nodes is specified and SplitGraph is enabled, only the graph that contains the root node is laid out by Arrange.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.SplitGraph Property

Gets or sets a value indicating whether unconnected subgraphs should be laid out independently from each other.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool SplitGraph { get; set; }
```

Visual Basic Copy Code

```
Public Property SplitGraph As Boolean
```

Property Value

true if unconnected graphs in the diagram should be laid out independently; otherwise, **false**. The default is **false**.

Remarks

If this property is set to true, each independent graph in the diagram is laid out in its own layout area. Otherwise all subgraphs are arranged in the same layout area. If a Root node is specified and **SplitGraph** is **true**, only the graph that contains the root node will be arranged.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.Stages Property

Gets or sets how many cool-down stages the algorithm should simulate.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int Stages { get; set; }
```

Visual Basic Copy Code

```
Public Property Stages As Integer
```

Property Value

An integer value specifying the number of temperature decreases the simulation should run. The default is 15.

Remarks

At each stage the algorithm evaluates different graph arrangements produced by modifying the best configuration from the previous stage. The modifications are done by shifting nodes by a distance that depends on the current temperature. The TemperatureScale property specifies how much the temperature decreases at each stage. IterationsPerStage specifies how many configurations are evaluated at each stage.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace
Progress Property

AnnealLayout.Statistics Property

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public AnnealLayoutStatistics Statistics { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Statistics As AnnealLayoutStatistics
```

Property Value

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout.TemperatureScale Property

Gets or sets how much the simulated temperature is decreased at each stage of the algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double TemperatureScale { get; set; }
```

Visual Basic Copy Code

```
Public Property TemperatureScale As Double
```

Property Value

A double multiplier applied to the simulated temperature at each stage of the algorithm. The default is 0.75.

Remarks

This value should be in the interval (0,1). The algorithm produces better results when running for many Stages with a gradual temperature decrease, that is, a **TemperatureScale** closer to 1.

See Also

AnnealLayout.WidthHeightRatio Property

Gets or sets what width/height ratio the layout area should have.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double WidthHeightRatio { get; set; }
```

Visual Basic Copy Code

```
Public Property WidthHeightRatio As Double
```

Property Value

A **double** value specifying the layout area width to height ratio. The default is 1.

Remarks

If the LayoutArea is not explicitly set, the algorithm will choose one suitable for the graph size. In that case the **WidthHeightRatio** property specifies how much larger the layout area width should be than its height.

See Also

AnnealLayout Members
AnnealLayout Class
MindFusion.Diagramming.Layout Namespace

AnnealLayout Events

Layout.LayoutComplete Event

Raised when a layout is finished arranging a graph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public event EventHandler LayoutComplete
```

Visual Basic Copy Code

```
Public Event LayoutComplete As EventHandler
```

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

CascadeLayout Class

Performs cascading layout on simple, connected graphs.

Syntax

C# Copy Code

```
public class CascadeLayout : Layout
```

Visual Basic Copy Code

```
Public Class CascadeLayout
    Inherits Layout
```

Remarks

CascadeLayout places nodes on a virtual grid and arranges links orthogonally, such that if the source graph is planar all links are guaranteed to have no more than two bends and will not intersect. By default the layout method arranges nodes in rows and link segments in columns; this can be changed by setting the Orientation property.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.CascadeLayout

See Also

CascadeLayout Members
MindFusion.Diagramming.Layout Namespace

CascadeLayout Members

Performs cascading layout on simple, connected graphs.

The following tables list the members exposed by the CascadeLayout type.

Public Constructors

	Name	Description
	CascadeLayout	Initializes a new instance of the **CascadeLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)

	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	Orientation	Gets or sets the orientation of the arranged graph.
	Padding	Gets or sets the spacing between nodes in adjacent layers.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

CascadeLayout Class
MindFusion.Diagramming.Layout Namespace

CascadeLayout Constructor

Initializes a new instance of the CascadeLayout class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public CascadeLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

CascadeLayout Class
MindFusion.Diagramming.Layout Namespace

CascadeLayout Properties

CascadeLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

CascadeLayout Members
CascadeLayout Class
MindFusion.Diagramming.Layout Namespace

CascadeLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Cascade layout".

See Also

CascadeLayout Members
CascadeLayout Class
MindFusion.Diagramming.Layout Namespace

CascadeLayout.Orientation Property

Gets or sets the orientation of the arranged graph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic Copy Code

```
Public Property Orientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default value is **Vertical**.

See Also

CascadeLayout Members
CascadeLayout Class
MindFusion.Diagramming.Layout Namespace

CascadeLayout.Padding Property

Gets or sets the spacing between nodes in adjacent layers.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double Padding { get; set; }
```

Visual Basic Copy Code

```
Public Property Padding As Double
```

Property Value

The default value is 15.

See Also

CascadeLayout Members
CascadeLayout Class
MindFusion.Diagramming.Layout Namespace

CascadeLayout Events

CircularLayout Class

Implements a circular graph layout algorithm. Nodes are distributed evenly on the circumference of a circle at positions that result in as few link crossing as possible.

Syntax

C# Copy Code

```
public class CircularLayout : Layout
```

Visual Basic Copy Code

```
Public Class CircularLayout
    Inherits Layout
```

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.CircularLayout

See Also

CircularLayout Members
MindFusion.Diagramming.Layout Namespace

CircularLayout Members

Implements a circular graph layout algorithm. Nodes are distributed evenly on the circumference of a circle at positions that result in as few link crossing as possible.

The following tables list the members exposed by the CircularLayout type.

Public Constructors

	Name	Description
	CircularLayout	Initializes a new instance of the **CircularLayout** class with the default settings.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	AutoRadiusFactor	Gets or sets a multiplier for the automatically calculated radius when Radius is set to 0.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)

	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	Radius	Gets or sets the radius of the layout circle.
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	SiftingRounds	Gets or sets the number of sifts to perform during the layout.
	SplitGraph	Gets or sets a value indicating whether to split the graph to its interconnected subgraphs and arrange the subgraphs independently or arrange the graph as a whole.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

CircularLayout Class
MindFusion.Diagramming.Layout Namespace

CircularLayout Constructor

Initializes a new instance of the CircularLayout class with the default settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public CircularLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

CircularLayout Class
MindFusion.Diagramming.Layout Namespace

CircularLayout Properties

CircularLayout.AutoRadiusFactor Property

Gets or sets a multiplier for the automatically calculated radius when Radius is set to 0.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double AutoRadiusFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoRadiusFactor As Double
```

Property Value

See Also

CircularLayout Members
CircularLayout Class
MindFusion.Diagramming.Layout Namespace

CircularLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

CircularLayout Members
CircularLayout Class
MindFusion.Diagramming.Layout Namespace

CircularLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Circular layout".

See Also

CircularLayout Members
CircularLayout Class
MindFusion.Diagramming.Layout Namespace

CircularLayout.Radius Property

Gets or sets the radius of the layout circle.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double Radius { get; set; }
```

Visual Basic Copy Code

```
Public Property Radius As Double
```

Property Value

A **double** value specifying the radius. The default value is 60.

Remarks

If this property is set to 0, the layout will try to find an appropriate radius depending on the number and size of the participating nodes.

See Also

CircularLayout Members
CircularLayout Class
MindFusion.Diagramming.Layout Namespace

CircularLayout.SiftingRounds Property

Gets or sets the number of sifts to perform during the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int SiftingRounds { get; set; }
```

Visual Basic Copy Code

```
Public Property SiftingRounds As Integer
```

Property Value

An integer value specifying the number of sifting rounds. The default value is 1.

Remarks

After calculating the initial placement of nodes using a fast but inaccurate layout function, the algorithm can run a few sifting rounds to reduce the link crossings, as set in this property.

See Also

CircularLayout Members
CircularLayout Class
MindFusion.Diagramming.Layout Namespace

CircularLayout.SplitGraph Property

Gets or sets a value indicating whether to split the graph to its interconnected subgraphs and arrange the subgraphs independently or arrange the graph as a whole.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool SplitGraph { get; set; }
```

Visual Basic Copy Code

```
Public Property SplitGraph As Boolean
```

Property Value

true if unconnected graphs in the diagram should be laid out independently; otherwise, **false**. The default is **true**.

See Also

CircularLayout Members
CircularLayout Class
MindFusion.Diagramming.Layout Namespace

CircularLayout Events

CompositeLayout Class

Implements a multi-level graph layout algorithm.

Syntax

C# Copy Code

```
public class CompositeLayout : Layout
```

Visual Basic Copy Code

```
Public Class CompositeLayout
    Inherits Layout
```

Remarks

CompositeLayout partitions the diagram into several subgraphs and applies the algorithm specified via the SubgraphLayout property on each part. If the part is a tree, it is arranged using the algorithm specified via the SubtreeLayout property, which is set to a radial TreeLayout instance by default. Finally the algorithm specified via MasterLayout is applied on the graph that represents the overall partition. By running the polynomial-complexity layout algorithms on small subgraphs, **CompositeLayout** is able to process a large graph much faster than if a layout algorithm is applied on the whole graph.

CompositeLayout can run on custom partitions specified as lists of nodes, or automatically partition the diagram via two automatic methods based on graph connectivity and graph path lengths. Set the PartitionMethod and CustomPartition properties to specify how the diagram should be partitioned.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.CompositeLayout

See Also

CompositeLayout Members
MindFusion.Diagramming.Layout Namespace

CompositeLayout Members

Implements a multi-level graph layout algorithm.

The following tables list the members exposed by the CompositeLayout type.

Public Constructors

	Name	Description
	CompositeLayout	Initializes a new instance of the **CompositeLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	BendLinks	Gets or sets a value indicating whether the links in the trees from the additional partitioning pass will bend to avoid crossing leaves in the same tree.

	CustomLayout	Gets or sets a custom layout method that should be invoked to arrange subgraphs.
	CustomPartition	Gets or sets the custom partition to use when PartitionMethod is set to Custom.
	DesiredSubgraphSize	Gets or sets the preferred number of nodes in the subgraphs allocated by the layout algorithm.
	EmphasizeTrees	Gets or sets a value indicating whether a cut-vertex between tree and non-tree subgraphs should be considered part of the tree.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	KeepVirtualItems	Gets or sets a value indicating whether to keep the intermediate virtual nodes and links.
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MasterLayout	Gets or sets a layout that should be applied globally to the diagram.
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	PartitionMethod	Gets or sets the partitioning method used to decompose the diagram into subgraphs.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	RotateSubgraphs	Gets or sets a value indicating whether subgraphs are rotated to minimize intersections with cross-partition links.
	SubgraphLayout	Gets or sets a layout that should be applied to general subgraphs.
	SubtreeLayout	Gets or sets a layout that should be applied to subgraphs representing trees.
	Tolerance	A multiplier applied to DesiredSubgraphSize to define the allowed subgraph size range.
	UseAdditionalPartitioning	Gets or sets a value indicating whether to use an additional partitioning pass on the result of the initial partitioning. This additional partitioning is simplified - it only looks for trees with depth 2.

Protected Properties

	Name	Description

	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout Constructor

Initializes a new instance of the CompositeLayout class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public CompositeLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout Properties

CompositeLayout.BendLinks Property

Gets or sets a value indicating whether the links in the trees from the additional partitioning pass will bend to avoid crossing leaves in the same tree.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool BendLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property BendLinks As Boolean
```

Property Value

true to bend links in the additional partitioning pass; otherwise, **false**. The default is **false**.

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout.CustomLayout Property

Gets or sets a custom layout method that should be invoked to arrange subgraphs.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public LayoutSubgraph CustomLayout { get; set; }
```

Visual Basic Copy Code

```
Public Property CustomLayout As LayoutSubgraph
```

Property Value

A LayoutSubgraph delegate object.

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout.CustomPartition Property

Gets or sets the custom partition to use when PartitionMethod is set to Custom.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public List<DiagramNodeCollection> CustomPartition { get; set; }
```

Visual Basic Copy Code

```
Public Property CustomPartition As List(Of DiagramNodeCollection)
```

Property Value

A list of DiagramNodeCollection objects that partitions the diagram into subgraphs.

Remarks

This value is used if PartitionMethod is set to Custom.

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout.DesiredSubgraphSize Property

Gets or sets the preferred number of nodes in the subgraphs allocated by the layout algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int DesiredSubgraphSize { get; set; }
```

Visual Basic Copy Code

```
Public Property DesiredSubgraphSize As Integer
```

Property Value

An integer value specifying the desired number of nodes in each subgraph.

Remarks

This property is considered by the automatic partitioning algorithm when generating subgraphs. Usually the subgraphs won't contain exactly as many nodes, but are within the Tolerance range around the property value.

See Also

CompositeLayout.EmphasizeTrees Property

Gets or sets a value indicating whether a cut-vertex between tree and non-tree subgraphs should be considered part of the tree.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool EmphasizeTrees { get; set; }
```

Visual Basic Copy Code

```
Public Property EmphasizeTrees As Boolean
```

Property Value

true to add cut-vertices to trees, otherwise **false**. The default value is **true**.

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace
SubtreeLayout Property

CompositeLayout.KeepVirtualItems Property

Gets or sets a value indicating whether to keep the intermediate virtual nodes and links.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool KeepVirtualItems { get; set; }
```

Visual Basic Copy Code

```
Public Property KeepVirtualItems As Boolean
```

Property Value

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout.MasterLayout Property

Gets or sets a layout that should be applied globally to the diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Layout MasterLayout { get; set; }
```

Visual Basic Copy Code

```
Public Property MasterLayout As Layout
```

Property Value

An instance of Layout-derived class. The default is AnnealLayout.

Remarks

Each subset in the layout partition is considered a node in a graph representing the diagram. **MasterLayout** is used to arrange that graph, while SubgraphLayout and SubtreeLayout are applied on the partition subsets. Since in the general case subsets will occupy areas of varying size, **MasterLayout** should be set to an instance of a layout class that can handle nodes of non-uniform dimensions.

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Composite layout".

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout.PartitionMethod Property

Gets or sets the partitioning method used to decompose the diagram into subgraphs.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public PartitionMethod PartitionMethod { get; set; }
```

Visual Basic

```
Public Property PartitionMethod As PartitionMethod
```

Property Value

A member of the PartitionMethod enumeration. The default value is **BiconnectedComponents**.

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout.RotateSubgraphs Property

Gets or sets a value indicating whether subgraphs are rotated to minimize intersections with cross-partition links.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public bool RotateSubgraphs { get; set; }
```

Visual Basic

```
Public Property RotateSubgraphs As Boolean
```

Property Value

true to rotate subgraphs, otherwise **false**. The default value is **false**.

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout.SubgraphLayout Property

Gets or sets a layout that should be applied to general subgraphs.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public Layout SubgraphLayout { get; set; }
```

Visual Basic

```
Public Property SubgraphLayout As Layout
```

Property Value

An instance of Layout-derived class. The default is CircularLayout.

Remarks

If the subgraph is a tree, CompositeLayout uses SubtreeLayout to arrange it.

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace
MasterLayout Property

CompositeLayout.SubtreeLayout Property

Gets or sets a layout that should be applied to subgraphs representing trees.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Layout SubtreeLayout { get; set; }
```

Visual Basic Copy Code

```
Public Property SubtreeLayout As Layout
```

Property Value

An instance of Layout-derived class. The default is radial TreeLayout.

Remarks

The trees isolated by the automatic partitioning methods are undirected, hence when using a custom tree layout you should use settings that ignore the direction of links. Alternatively, use custom partitioning to ensure subgraphs are directed trees, i.e. the links in paths from the root to each node follow the same direction.

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace
MasterLayout Property
SubgraphLayout Property
PartitionMethod Property

CompositeLayout.Tolerance Property

A multiplier applied to DesiredSubgraphSize to define the allowed subgraph size range.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double Tolerance { get; set; }
```

Visual Basic Copy Code

```
Public Property Tolerance As Double
```

Property Value

A **double** value specifying the tolerance range. The default is 0.2.

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout.UseAdditionalPartitioning Property

Gets or sets a value indicating whether to use an additional partitioning pass on the result of the initial partitioning. This additional partitioning is simplified - it only looks for trees with depth 2.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool UseAdditionalPartitioning { get; set; }
```

Visual Basic Copy Code

```
Public Property UseAdditionalPartitioning As Boolean
```

Property Value

true to use second partitioning pass; otherwise, **false**. The default is **false**.

See Also

CompositeLayout Members
CompositeLayout Class
MindFusion.Diagramming.Layout Namespace

CompositeLayout Events

DecisionLayout Class

Implements a flowchart layout algorithm.

Syntax

C# Copy Code

```
public class DecisionLayout : Layout
```

Visual Basic Copy Code

```
Public Class DecisionLayout
    Inherits Layout
```

Remarks

This algorithm arranges simple flowcharts consisting of decision boxes with up to three outgoing links per node and activity boxes with a single outgoing link per node. The nodes are arranged in columns and rows, whose distance depends on the HorizontalPadding and VerticalPadding property values. When links share the same row or column, they are placed at a distance specified via LinkPadding.

The layout arranges nodes recursively starting from StartNode. If **StartNode** is not specified, the algorithm selects the root of the deepest branch of the graph's spanning tree as start node.

The overall orientation of the layout can be set via the Orientation property. There is no distinction made between yes/no links, and the node sides by which links are aligned depend only on the order of links in **OutgoingLinks** collection. If you have a preferred position for links depending on their yes/no type, make sure the **OutgoingLinks** collections of nodes are sorted by links' logical values.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.DecisionLayout

See Also

DecisionLayout Members
MindFusion.Diagramming.Layout Namespace
FlowchartLayout Class

DecisionLayout Members

Implements a flowchart layout algorithm.

The following tables list the members exposed by the DecisionLayout type.

Public Constructors

	Name	Description
	DecisionLayout	Initializes a new instance of the **DecisionLayout** class with the default settings.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	HorizontalPadding	Gets or sets the horizontal distance between adjacent nodes.
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	LinkPadding	Gets or sets the distance between parallel links.
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	Orientation	Gets or sets the orientation of the arranged diagram.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	StartNode	Gets or sets the initial (top-left) node in the arranged diagram.
	VerticalPadding	Gets or sets the vertical distance between adjacent nodes.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

DecisionLayout Class
MindFusion.Diagramming.Layout Namespace

DecisionLayout Constructor

Initializes a new instance of the DecisionLayout class with the default settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public DecisionLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DecisionLayout Class
MindFusion.Diagramming.Layout Namespace

DecisionLayout Properties

DecisionLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

DecisionLayout Members
DecisionLayout Class
MindFusion.Diagramming.Layout Namespace

DecisionLayout.HorizontalPadding Property

Gets or sets the horizontal distance between adjacent nodes.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double HorizontalPadding { get; set; }
```

Visual Basic Copy Code

```
Public Property HorizontalPadding As Double
```

Property Value

A **double** value specifying the distance between adjacent nodes in same row. The default value is 15.

See Also

DecisionLayout Members
DecisionLayout Class
MindFusion.Diagramming.Layout Namespace

DecisionLayout.LinkPadding Property

Gets or sets the distance between parallel links.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double LinkPadding { get; set; }
```

Visual Basic

```
Public Property LinkPadding As Double
```

Property Value

A **double** value specifying the distance between links that share the same row or column. The default value is 2.

See Also

DecisionLayout Members
DecisionLayout Class
MindFusion.Diagramming.Layout Namespace

DecisionLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public override string Name { get; }
```

Visual Basic

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Decision layout".

See Also

DecisionLayout Members
DecisionLayout Class
MindFusion.Diagramming.Layout Namespace

DecisionLayout.Orientation Property

Gets or sets the orientation of the arranged diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public Orientation Orientation { get; set; }
```

Visual Basic

```
Public Property Orientation As Orientation
```

Property Value

A member of the Orientation enumeration specifying the overall layout orientation. The default value is **Vertical**.

See Also

DecisionLayout Members
DecisionLayout Class
MindFusion.Diagramming.Layout Namespace

DecisionLayout.StartNode Property

Gets or sets the initial (top-left) node in the arranged diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public DiagramNode StartNode { get; set; }
```

Visual Basic Copy Code

```
Public Property StartNode As DiagramNode
```

Property Value

A DiagramNode instance specifying the start node.

Remarks

If this property is **null**, the layout will pick-up a start node automatically.

See Also

DecisionLayout Members
DecisionLayout Class
MindFusion.Diagramming.Layout Namespace

DecisionLayout.VerticalPadding Property

Gets or sets the vertical distance between adjacent nodes.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double VerticalPadding { get; set; }
```

Visual Basic Copy Code

```
Public Property VerticalPadding As Double
```

Property Value

A **double** value specifying the distance between adjacent nodes in same column. The default value is 15.

See Also

DecisionLayout Members
DecisionLayout Class
MindFusion.Diagramming.Layout Namespace

DecisionLayout Events

Direction Enumeration

Specifies a layout direction.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum Direction
```

Visual Basic Copy Code

```
Public Enum Direction
```

Members

	Member name	Description
	Straight	Indicates either left-to-right or top-to-bottom direction.
	Reversed	Indicates either right-to-left or bottom-to-top direction.

See Also

MindFusion.Diagramming.Layout Namespace

FlowchartLayout Class

FlowchartLayout can be used to arrange flowcharts representing program source code.

Syntax

C# Copy Code

```
public class FlowchartLayout : Layout
```

Visual Basic Copy Code

```
Public Class FlowchartLayout
    Inherits Layout
```

Remarks

The algorithm recognizes program code -like patterns in the graph, such as loops, sequences and if/switch branchings, and arranges them recursively. **FlowchartLayout** could be used to arrange other types of graphs as well, though with some restrictions. For example it treats all back links as loops in the code, and expects that they are nested - loop links starting closer to the stop node should end closer to the start node. Another similar restriction is that there shouldn't be any cross-links that connect different branches of a decision sub-graph.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.FlowchartLayout

See Also

FlowchartLayout Members
MindFusion.Diagramming.Layout Namespace
DecisionLayout Class

FlowchartLayout Members

FlowchartLayout can be used to arrange flowcharts representing program source code.

The following tables list the members exposed by the FlowchartLayout type.

Public Constructors

	Name	Description
	FlowchartLayout	Initializes a new instance of the **FlowchartLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	BranchPadding	Gets or sets how much space to leave between adjacent decision branches in the flowchart.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	IgnoreRepeatingLinks	Gets or sets a value indicating whether to ignore repeating links.
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	LinkPadding	Gets or sets how much space to leave between adjacent back links designating nested loops.
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	NodeDistance	Gets or sets the distance between consecutive nodes in the flowchart.
	Orientation	Gets or sets the overall orientation of the arranged diagram.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	SplitGraph	Gets or sets a value indicating whether unconnected subgraphs should be laid out independently from each other.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

FlowchartLayout Class
MindFusion.Diagramming.Layout Namespace

FlowchartLayout Constructor

Initializes a new instance of the FlowchartLayout class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public FlowchartLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

FlowchartLayout Class
MindFusion.Diagramming.Layout Namespace

FlowchartLayout Properties

FlowchartLayout.BranchPadding Property

Gets or sets how much space to leave between adjacent decision branches in the flowchart.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double BranchPadding { get; set; }
```

Visual Basic Copy Code

```
Public Property BranchPadding As Double
```

Property Value

A **Double** value specifying the distance between adjacent branches.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Diagramming.Layout Namespace

FlowchartLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Diagramming.Layout Namespace

FlowchartLayout.IgnoreRepeatingLinks Property

Gets or sets a value indicating whether to ignore repeating links.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool IgnoreRepeatingLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property IgnoreRepeatingLinks As Boolean
```

Property Value

true to ignore repeating links, otherwise **false**. The default value is **false**.

Remarks

If there are many links connecting the same pair of nodes in the same direction, the analysis of the inherent flowchart structure might fail. In such cases, set this property to **true**.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Diagramming.Layout Namespace

FlowchartLayout.LinkPadding Property

Gets or sets how much space to leave between adjacent back links designating nested loops.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double LinkPadding { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkPadding As Double
```

Property Value

A **double** value specifying the distance between adjacent back links.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Diagramming.Layout Namespace

FlowchartLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Flowchart layout".

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Diagramming.Layout Namespace

FlowchartLayout.NodeDistance Property

Gets or sets the distance between consecutive nodes in the flowchart.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double NodeDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeDistance As Double
```

Property Value

A **double** value specifying the distance between neighbor nodes in the same sequence.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Diagramming.Layout Namespace

FlowchartLayout.Orientation Property

Gets or sets the overall orientation of the arranged diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic Copy Code

```
Public Property Orientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default value is **Vertical**.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Diagramming.Layout Namespace

FlowchartLayout.SplitGraph Property

Gets or sets a value indicating whether unconnected subgraphs should be laid out independently from each other.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool SplitGraph { get; set; }
```

Visual Basic Copy Code

```
Public Property SplitGraph As Boolean
```

Property Value

true to arrange each subgraph on its own, otherwise **false**. The default value is **true**.

Remarks

If this property is set to **true**, each independent graph in the diagram is laid out in its own layout area. Otherwise all subgraphs are arranged in the same layout area.

See Also

FlowchartLayout Members
FlowchartLayout Class
MindFusion.Diagramming.Layout Namespace

FlowchartLayout Events

FractalLayout Class

A tree layout algorithm that places child nodes symmetrically around their parent node.

Syntax

C# Copy Code

```
public class FractalLayout : Layout
```

Visual Basic Copy Code

```
Public Class FractalLayout
    Inherits Layout
```

Remarks

Nodes at the lowest level are arranged directly in a circle around their parent. At the upper level, the already arranged nodes form branches that are arranged in a circle around the new parent node. The algorithm is recursively repeated till the highest level is reached. If nodes in the tree have uniform number of children, the end result has fractal-like appearance (subsets of the graph look like scaled-down copies of the whole graph).

You can choose which node should be displayed at the center of the topmost circle by setting the Root property. If it is not specified, the algorithm automatically selects a root that leads to more balanced distribution of nodes.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.FractalLayout

See Also

FractalLayout Members
MindFusion.Diagramming.Layout Namespace

FractalLayout Members

A tree layout algorithm that places child nodes symmetrically around their parent node.

The following tables list the members exposed by the FractalLayout type.

Public Constructors

	Name	Description
	FractalLayout	Initializes a new instance of the **FractalLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	Root	Gets or sets the tree root.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

FractalLayout Class
MindFusion.Diagramming.Layout Namespace

FractalLayout Constructor

Initializes a new instance of the FractalLayout class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public FractalLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

FractalLayout Class
MindFusion.Diagramming.Layout Namespace

FractalLayout Properties

FractalLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

See Also

FractalLayout Members
FractalLayout Class
MindFusion.Diagramming.Layout Namespace

FractalLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Fractal layout".

See Also

FractalLayout Members
FractalLayout Class
MindFusion.Diagramming.Layout Namespace

FractalLayout.Root Property

Gets or sets the tree root.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public DiagramNode Root { get; set; }
```

Visual Basic Copy Code

```
Public Property Root As DiagramNode
```

Property Value

A DiagramNode that should be placed at the center of the arranged tree. The default value is **null**.

Remarks

If there is no root specified, the algorithm automatically selects one that will lead to more balanced branch sizes.

See Also

FractalLayout Members
FractalLayout Class
MindFusion.Diagramming.Layout Namespace

FractalLayout Events

GridLayout Class

Implements a grid graph layout algorithm.

Syntax

C# Copy Code

```
public class GridLayout : Layout
```

Visual Basic Copy Code

```
Public Class GridLayout
    Inherits Layout
```

Remarks

GridLayout arranges diagram nodes in a grid, keeping connected nodes close together. The algorithm strives to achieve a small number of link crossings. It is based on an iterative process whose initial steps shuffle the grid nodes randomly. That can lead to very different results each time the algorithm is run.

To apply the layout to a diagram, create a **GridLayout** instance, set its members and invoke the Arrange method. Arrange lays out all subgraphs in a diagram, unless the Root property is set.

The distance between points in the layout grid can be changed through GridSize. Nodes to be placed at the upper-left and lower-right corners of the grid can be assigned to StartNode and EndNode.

Anchoring specifies whether the layout algorithm must attach links to the anchor points of graphs nodes. The grid layout algorithm produces best results if it can place link ends at arbitrary positions over nodes' outlines, that is, when using the Ignore value. If your application logic requires links to be always attached to anchor points, use the other two values. In such cases **Reassign** would yield better results.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.GridLayout

See Also

GridLayout Members
MindFusion.Diagramming.Layout Namespace

GridLayout Members

Implements a grid graph layout algorithm.

The following tables list the members exposed by the GridLayout type.

Public Constructors

	Name	Description
	GridLayout	Initializes a new instance of the **GridLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)

	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	EndNode	Gets or sets the node that is placed at the lower right corner of the grid.
	GridSize	Gets or sets the desired distance between adjacent grid points.
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	Iterations	Gets or sets for how many iterations to run the grid layout algorithm.
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RandomSeed	Gets or sets the randomization seed to use for generating random node positions.
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	Root	Gets or sets a node indicating which subgraph of the diagram should be laid out.
	SplitGraph	Gets or sets a value indicating whether to split the graph to its interconnected subgraphs and arrange the subgraphs independently or arrange the graph as a whole.
	StartNode	Gets or sets the node that is placed at the upper left corner of the grid.
	TimeLimit	Gets or sets a time-out value for the path-finding part of the grid layout algorithm.
	UseLongestPath	Gets or sets a value indicating whether the algorithm should arrange nodes around the longest path in the graph.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description

	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout Constructor

Initializes a new instance of the GridLayout class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public GridLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout Properties

GridLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(10, 10).

See Also

GridLayout Members
GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout.EndNode Property

Gets or sets the node that is placed at the lower right corner of the grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public DiagramNode EndNode { get; set; }
```

Visual Basic

```
Public Property EndNode As DiagramNode
```

Property Value

An instance of the DiagramNode class. Default value is **null** (**Nothing** in Visual Basic).

Remarks

If you assign a node to that property, then StartNode cannot be **null** and must be set too. GridLayout usually produces better layouts if both **StartNode** and **EndNode** are **null** references; in such a case the algorithm automatically selects suitable nodes.

See Also

GridLayout Members
GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout.GridSize Property

Gets or sets the desired distance between adjacent grid points.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public double GridSize { get; set; }
```

Visual Basic

```
Public Property GridSize As Double
```

Property Value

A **double** number specifying the grid size. The default is 25.

Remarks

When placing a node at a grid position, the node center point is aligned to the grid point at that position.

See Also

GridLayout Members
GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout.Iterations Property

Gets or sets for how many iterations to run the grid layout algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int Iterations { get; set; }
```

Visual Basic Copy Code

```
Public Property Iterations As Integer
```

Property Value

An integer number specifying the number of iterations. The default is 3000.

Remarks

More iterations would yield better layouts but require more time to complete.

See Also

GridLayout Members
GridLayout Class
MindFusion.Diagramming.Layout Namespace
Progress Property

GridLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Grid layout".

See Also

GridLayout Members
GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout.RandomSeed Property

Gets or sets the randomization seed to use for generating random node positions.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int RandomSeed { get; set; }
```

Visual Basic Copy Code

```
Public Property RandomSeed As Integer
```

Property Value

See Also

GridLayout Members
GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout.Root Property

Gets or sets a node indicating which subgraph of the diagram should be laid out.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public DiagramNode Root { get; set; }
```

Visual Basic Copy Code

```
Public Property Root As DiagramNode
```

Property Value

An instance of the DiagramNode class. The default is a **null** reference (**Nothing** in Visual Basic).

Remarks

All links and nodes that can be reached from the root by following recursively its links form a graph. If the root is specified, only the graph which it defines is laid out by Arrange.

See Also

GridLayout Members
GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout.SplitGraph Property

Gets or sets a value indicating whether to split the graph to its interconnected subgraphs and arrange the subgraphs independently or arrange the graph as a whole.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool SplitGraph { get; set; }
```

Visual Basic Copy Code

```
Public Property SplitGraph As Boolean
```

Property Value

true to split the graph; otherwise, **false**.

See Also

GridLayout Members
GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout.StartNode Property

Gets or sets the node that is placed at the upper left corner of the grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public DiagramNode StartNode { get; set; }
```

Visual Basic Copy Code

```
Public Property StartNode As DiagramNode
```

Property Value

An instance of the DiagramNode class. The default is **null** (**Nothing** in Visual Basic).

Remarks

If you assign a node to that property, then EndNode cannot be **null** and must be set too. GridLayout usually produces better layouts if both **StartNode** and **EndNode** are **null** references; in such a case the algorithm automatically selects suitable nodes.

See Also

GridLayout Members
GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout.TimeLimit Property

Gets or sets a time-out value for the path-finding part of the grid layout algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public long TimeLimit { get; set; }
```

Visual Basic Copy Code

```
Public Property TimeLimit As Long
```

Property Value

An integer value specifying time-out in milliseconds. The default is 10000 (10 seconds).

Remarks

The grid layout algorithm arranges graph nodes and links along the longest path in the graph. Large graphs with many links could contain hundreds of thousands of possible paths, so finding the longest one would be very time-consuming. When the time-out set with this property occurs, the path-finding routine stops and the layout algorithm uses the longest path known at that moment. Usually, for graphs containing several hundreds of nodes, 1-2 seconds are enough to find a path long enough to produce a nice layout.

See Also

GridLayout Members
GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout.UseLongestPath Property

Gets or sets a value indicating whether the algorithm should arrange nodes around the longest path in the graph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool UseLongestPath { get; set; }
```

Visual Basic Copy Code

```
Public Property UseLongestPath As Boolean
```

Property Value

true to look for the longest path in the graph and arrange nodes around it; otherwise, **false**.

Remarks

Initially, the algorithm places nodes and links from the longest path in the graph at the grid diagonal, and arranges the rest of the nodes around the diagonal. Large graphs with many links could contain hundreds of thousands of possible paths, so finding the longest one would be very time-consuming. Disabling this property will let you skip the path-finding part of the algorithm, which will make it work much faster.

See Also

GridLayout Members
GridLayout Class
MindFusion.Diagramming.Layout Namespace

GridLayout Events

HierarchicalLayout Class

Performs hierarchical layout on simple, connected, level-planar graphs.

Syntax

C# Copy Code

```
public class HierarchicalLayout : Layout
```

Visual Basic Copy Code

```
Public Class HierarchicalLayout
    Inherits Layout
```

Remarks

HierarchicalLayout places nodes on user-defined levels, such that if the source graph is level-planar, all links are guaranteed to have a single segment and will not intersect. A graph is considered level-planar if it is possible to draw it without link intersections when its nodes are positioned on predefined levels. The layout method requires that for each node LayoutTraits contains a HierarchicalLayoutTraits.Level entry specifying the level, and no two connected nodes must be on the same level.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.HierarchicalLayout

See Also

HierarchicalLayout Members
MindFusion.Diagramming.Layout Namespace

HierarchicalLayout Members

Performs hierarchical layout on simple, connected, level-planar graphs.

The following tables list the members exposed by the HierarchicalLayout type.

Public Constructors

	Name	Description
	HierarchicalLayout	Initializes a new instance of the **HierarchicalLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	LevelDistance	Gets or sets the distance between adjacent levels in the hierarchical layout.
	LinkBendFactor	Gets or sets the bending factor of links.
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)

	Name	Gets a user-friendly name of the layout.
	Orientation	Gets or sets the orientation of the arranged graph.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

HierarchicalLayout Class
MindFusion.Diagramming.Layout Namespace

HierarchicalLayout Constructor

Initializes a new instance of the HierarchicalLayout class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public HierarchicalLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

HierarchicalLayout Class
MindFusion.Diagramming.Layout Namespace

HierarchicalLayout Properties

HierarchicalLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

HierarchicalLayout Members
HierarchicalLayout Class
MindFusion.Diagramming.Layout Namespace

HierarchicalLayout.LevelDistance Property

Gets or sets the distance between adjacent levels in the hierarchical layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double LevelDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property LevelDistance As Double
```

Property Value

The default value is 40.

See Also

HierarchicalLayout Members
HierarchicalLayout Class
MindFusion.Diagramming.Layout Namespace

HierarchicalLayout.LinkBendFactor Property

Gets or sets the bending factor of links.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double LinkBendFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkBendFactor As Double
```

Property Value

The default value is 1.

Remarks

A value of 1 indicates that the links should not bend. A value less than 1 indicates that the links should bend towards the geometric center of the graph. A value greater than 1 indicates that the links should bend outwards.

See Also

HierarchicalLayout Members
HierarchicalLayout Class
MindFusion.Diagramming.Layout Namespace

HierarchicalLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Hierarchical layout".

See Also

HierarchicalLayout Members
HierarchicalLayout Class
MindFusion.Diagramming.Layout Namespace

HierarchicalLayout.Orientation Property

Gets or sets the orientation of the arranged graph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic Copy Code

```
Public Property Orientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default value is **Vertical**.

See Also

HierarchicalLayout Members
HierarchicalLayout Class
MindFusion.Diagramming.Layout Namespace

HierarchicalLayout Events

ILayout Interface

Common interface for all MvcDiagram layout classes.

Syntax

C# Copy Code

```
public interface ILayout
```

Visual Basic Copy Code

```
Public Interface ILayout
```

Inheritors

MindFusion.Diagramming.Layout.Layout

See Also

ILayout Members
MindFusion.Diagramming.Layout Namespace

ILayout Members

Common interface for all MvcDiagram layout classes.

The following tables list the members exposed by the ILayout type.

Public Properties

	Name	Description
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout.
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout.

Public Methods

	Name	Description

	Arrange	Overloaded.
	ArrangeAnimated	Overloaded.
	ArrangeAsync	Overloaded.

Public Events

	Name	Description
	LayoutComplete	

See Also

ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout Methods

ILayout.Arrange Method

Overload List

Name	Description
ILayout.Arrange (Diagram)	Applies the layout to the specified Diagram instance.
ILayout.Arrange (Diagram, DiagramItemCollection)	
ILayout.Arrange (Diagram, DiagramItemCollection, Double, Double)	

See Also

ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout.Arrange Method (Diagram)

Applies the layout to the specified Diagram instance.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
bool Arrange (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Function Arrange( _
    diagram As Diagram _
) As Boolean
```

Parameters

diagram
The Diagram that will be arranged.

Return Value

Returns whether the arrangement was successful.

See Also

Arrange Method Overload List
ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout.Arrange Method (Diagram, DiagramItemCollection)

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
bool Arrange (
    Diagram diagram,
    DiagramItemCollection items
)
```

Visual Basic Copy Code

```
Function Arrange( _
    diagram As Diagram, _
    items As DiagramItemCollection _
) As Boolean
```

Parameters

diagram

items

Return Value

See Also

Arrange Method Overload List
ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout.Arrange Method (Diagram, DiagramItemCollection, Double, Double)

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
bool Arrange (
    Diagram diagram,
    DiagramItemCollection items,
    double dx,
    double dy
)
```

Visual Basic

```
Function Arrange( _
    diagram As Diagram, _
    items As DiagramItemCollection, _
    dx As Double, _
    dy As Double _
) As Boolean
```

Parameters

diagram

items

dx

dy

Return Value

See Also

Arrange Method Overload List
ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout.ArrangeAnimated Method

Overload List

Name	Description
ILayout.ArrangeAnimated (Diagram, DiagramItemCollection, Int32, AnimationType, EasingType, Double, Double)	
ILayout.ArrangeAnimated (Diagram, Int32, AnimationType, EasingType)	

See Also

ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout.ArrangeAnimated Method (Diagram, DiagramItemCollection, Int32, AnimationType, EasingType, Double, Double)

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
bool ArrangeAnimated (
    Diagram diagram,
    DiagramItemCollection items,
    int duration,
    AnimationType animationType,
    EasingType easingType,
    double dx,
```

```
    double dy
)
```

Visual Basic Copy Code

```
Function ArrangeAnimated( _
    diagram As Diagram, _
    items As DiagramItemCollection, _
    duration As Integer, _
    animationType As AnimationType, _
    easingType As EasingType, _
    dx As Double, _
    dy As Double _
) As Boolean
```

Parameters

diagram

items

duration

animationType

easingType

dx

dy

Return Value

See Also

ArrangeAnimated Method Overload List
ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout.ArrangeAnimated Method (Diagram, Int32, AnimationType, EasingType)

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
bool ArrangeAnimated (
    Diagram diagram,
    int duration,
    AnimationType animationType,
    EasingType easingType
)
```

Visual Basic Copy Code

```
Function ArrangeAnimated( _
    diagram As Diagram, _
    duration As Integer, _
    animationType As AnimationType, _
    easingType As EasingType _
) As Boolean
```

Parameters

diagram

duration

animationType

easingType

Return Value

See Also

ArrangeAnimated Method Overload List
ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout.ArrangeAsync Method

Overload List

Name	Description
ILayout.ArrangeAsync (Diagram)	
ILayout.ArrangeAsync (Diagram, DiagramItemCollection)	
ILayout.ArrangeAsync (Diagram, DiagramItemCollection, Double, Double)	

See Also

ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout.ArrangeAsync Method (Diagram)

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
Task<Boolean> ArrangeAsync (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Function ArrangeAsync( _
    diagram As Diagram _
) As Task(Of Boolean)
```

Parameters

diagram

Return Value

See Also

ArrangeAsync Method Overload List
ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout.ArrangeAsync Method (Diagram, DiagramItemCollection)

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
Task<Boolean> ArrangeAsync (
    Diagram diagram,
    DiagramItemCollection items
)
```

Visual Basic Copy Code

```
Function ArrangeAsync( _
    diagram As Diagram, _
    items As DiagramItemCollection _
) As Task(Of Boolean)
```

Parameters

diagram

items

Return Value

See Also

ArrangeAsync Method Overload List
ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout.ArrangeAsync Method (Diagram, DiagramItemCollection, Double, Double)

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
Task<Boolean> ArrangeAsync (
    Diagram diagram,
    DiagramItemCollection items,
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Function ArrangeAsync( _
    diagram As Diagram, _
    items As DiagramItemCollection, _
    dx As Double, _
    dy As Double _
) As Task(Of Boolean)
```

Parameters

diagram

items

dx

dy

Return Value

See Also

ArrangeAsync Method Overload List
ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout Properties

ILayout.LayoutLink Property

Gets or sets a delegate, which is invoked when the position of a link is changed by the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
LayoutLink LayoutLink { get; set; }
```

Visual Basic Copy Code

```
Property LayoutLink As LayoutLink
```

Property Value

An instance of the LayoutLink delegate.

See Also

ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ILayout.LayoutNode Property

Gets or sets a delegate, which is invoked when the position of a node is changed by the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
LayoutNode LayoutNode { get; set; }
```

Visual Basic Copy Code

```
Property LayoutNode As LayoutNode
```

Property Value

An instance of the **LayoutNode** delegate.

See Also

ILayout Members
ILayout Interface

MindFusion.Diagramming.Layout Namespace

ILayout Events

ILayout.LayoutComplete Event

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
event EventHandler LayoutComplete
```

Visual Basic Copy Code

```
Event LayoutComplete As EventHandler
```

See Also

ILayout Members
ILayout Interface
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid Interface

Represents an object which can be used by a SwimlaneLayout to adjust the corresponding swimlane grid.

Syntax

C# Copy Code

```
public interface ISwimlaneGrid
```

Visual Basic Copy Code

```
Public Interface ISwimlaneGrid
```

Inheritors

MindFusion.Diagramming.Layout.LaneGrid

See Also

ISwimlaneGrid Members
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid Members

Represents an object which can be used by a SwimlaneLayout to adjust the corresponding swimlane grid.

The following tables list the members exposed by the ISwimlaneGrid type.

Public Methods

	Name	Description
	GetColumnCount	Gets the number of columns in the grid.

	GetColumnWidth	Gets the the width of the specified column in the swimlane grid.
	GetRowCount	Gets the number of rows in the swimlane grid.
	GetRowHeight	Gets the the height of the specified row in the swimlane grid.
	SetColumnCount	Sets the number of columns in the swimlane grid.
	SetColumnWidth	Sets the the width of the specified column in the swimlane grid.
	SetLeftMargin	Sets the left margin of the swimlane grid.
	SetRowCount	Sets the number of rows in the swimlane grid.
	SetRowHeight	Sets the the height of the specified row in the swimlane grid.
	SetTopMargin	Sets the top margin of the swimlane grid.

See Also

ISwimlaneGrid Interface
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid Methods

ISwimlaneGrid.GetColumnCount Method

Gets the number of columns in the grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
int GetColumnCount ()
```

Visual Basic Copy Code

```
Function GetColumnCount () As Integer
```

Return Value

The number of columns in the grid.

See Also

ISwimlaneGrid Members
ISwimlaneGrid Interface
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid.GetColumnWidth Method

Gets the the width of the specified column in the swimlane grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
double GetColumnWidth (
    int index
```

```
)
```

Visual Basic Copy Code

```
Function GetColumnWidth( _
    index As Integer _
) As Double
```

Parameters

index

The zero-based index of the column.

Return Value

The width of the column.

See Also

ISwimlaneGrid Members
ISwimlaneGrid Interface
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid.GetRowCount Method

Gets the number of rows in the swimlane grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
int GetRowCount ()
```

Visual Basic Copy Code

```
Function GetRowCount () As Integer
```

Return Value

The number of rows in the grid.

See Also

ISwimlaneGrid Members
ISwimlaneGrid Interface
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid.GetRowHeight Method

Gets the the height of the specified row in the swimlane grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
double GetRowHeight (
    int index
)
```

Visual Basic Copy Code

```
Function GetRowHeight( _
    index As Integer _
) As Double
```

Parameters

index
The zero-based index of the row.

Return Value

The height of the row.

See Also

ISwimlaneGrid Members
ISwimlaneGrid Interface
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid.SetColumnCount Method

Sets the number of columns in the swimlane grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
void SetColumnCount (
    int value
)
```

Visual Basic Copy Code

```
Sub SetColumnCount( _
    value As Integer _
)
```

Parameters

value
A non-negative integer value specifying the number of columns.

See Also

ISwimlaneGrid Members
ISwimlaneGrid Interface
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid.SetColumnWidth Method

Sets the the width of the specified column in the swimlane grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
void SetColumnWidth (
    int index,
    double value
)
```

Visual Basic

```
Sub SetColumnWidth( _
    index As Integer, _
    value As Double _
)
```

Parameters

index

The zero-based index of the column.

value

The new width of the column.

See Also

ISwimlaneGrid Members
ISwimlaneGrid Interface
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid.SetLeftMargin Method

Sets the left margin of the swimlane grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
void SetLeftMargin (
    double value
)
```

Visual Basic

```
Sub SetLeftMargin( _
    value As Double _
)
```

Parameters

value

The distance between the left side of the grid and the diagram bounds.

See Also

ISwimlaneGrid Members
ISwimlaneGrid Interface
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid.SetRowCount Method

Sets the number of rows in the swimlane grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
void SetRowCount (
    int value
)
```

Visual Basic Copy Code

```
Sub SetRowCount( _
    value As Integer _
)
```

Parameters

value
A non-negative integer value specifying the number of rows.

See Also

ISwimlaneGrid Members
ISwimlaneGrid Interface
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid.SetRowHeight Method

Sets the the height of the specified row in the swimlane grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
void SetRowHeight (
    int index,
    double value
)
```

Visual Basic Copy Code

```
Sub SetRowHeight( _
    index As Integer, _
    value As Double _
)
```

Parameters

index
The zero-based index of the row.

value
The new height of the row.

See Also

ISwimlaneGrid Members
ISwimlaneGrid Interface
MindFusion.Diagramming.Layout Namespace

ISwimlaneGrid.SetTopMargin Method

Sets the top margin of the swimlane grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
void SetTopMargin (
    double value
)
```

Visual Basic Copy Code

```
Sub SetTopMargin( _
    value As Double _
)
```

Parameters

value
The distance between the top side of the grid and the diagram bounds.

See Also

ISwimlaneGrid Members
ISwimlaneGrid Interface
MindFusion.Diagramming.Layout Namespace

LaneGrid Class

Represents an ISwimlaneGrid which operates on a MindFusion.Diagramming.Lanes.Grid object.

Syntax

C# Copy Code

```
public class LaneGrid : ISwimlaneGrid
```

Visual Basic Copy Code

```
Public Class LaneGrid
    Implements ISwimlaneGrid
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Layout.LaneGrid

See Also

LaneGrid Members
MindFusion.Diagramming.Layout Namespace

LaneGrid Members

Represents an ISwimlaneGrid which operates on a MindFusion.Diagramming.Lanes.Grid object.

The following tables list the members exposed by the LaneGrid type.

Public Constructors

	Name	Description
	LaneGrid	Initializes a new instance of the **LaneGrid** class.

Public Methods

	Name	Description
	GetColumnCount	ISwimlaneGrid.GetColumnCount implementation.

	GetColumnWidth	ISwimlaneGrid.GetColumnCount implementation.
	GetRowCount	ISwimlaneGrid.GetRowCount implementation.
	GetRowHeight	ISwimlaneGrid.GetRowHeight implementation.
	SetColumnCount	ISwimlaneGrid.SetColumnCount implementation.
	SetColumnWidth	ISwimlaneGrid.SetColumnCount implementation.
	SetLeftMargin	ISwimlaneGrid.SetLeftMargin implementation.
	SetRowCount	ISwimlaneGrid.SetRowCount implementation.
	SetRowHeight	ISwimlaneGrid.SetRowHeight implementation.
	SetTopMargin	ISwimlaneGrid.SetTopMargin implementation.

See Also

LaneGrid Class
MindFusion.Diagramming.Layout Namespace

LaneGrid Constructor

Initializes a new instance of the LaneGrid class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public LaneGrid (
    Grid grid
)
```

Visual Basic Copy Code

```
Public New ( _
    grid As Grid _
)
```

Parameters

grid

See Also

LaneGrid Class
MindFusion.Diagramming.Layout Namespace

LaneGrid Methods

LaneGrid.GetColumnCount Method

ISwimlaneGrid.GetColumnCount implementation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int GetColumnCount ()
```

Visual Basic Copy Code

```
Public Function GetColumnCount () As Integer
```

Return Value

See Also

LaneGrid Members
LaneGrid Class
MindFusion.Diagramming.Layout Namespace

LaneGrid.GetColumnWidth Method

ISwimlaneGrid.GetColumnCount implementation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double GetColumnWidth (
    int index
)
```

Visual Basic Copy Code

```
Public Function GetColumnWidth( _
    index As Integer _
) As Double
```

Parameters

index

Return Value

See Also

LaneGrid Members
LaneGrid Class
MindFusion.Diagramming.Layout Namespace

LaneGrid.GetRowCount Method

ISwimlaneGrid.GetRowCount implementation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int GetRowCount ()
```

Visual Basic Copy Code

```
Public Function GetRowCount () As Integer
```

Return Value

See Also

LaneGrid Members
LaneGrid Class
MindFusion.Diagramming.Layout Namespace

LaneGrid.GetRowHeight Method

ISwimlaneGrid.GetRowHeight implementation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double GetRowHeight (
    int index
)
```

Visual Basic Copy Code

```
Public Function GetRowHeight( _
    index As Integer _
) As Double
```

Parameters

index

Return Value

See Also

LaneGrid Members
LaneGrid Class
MindFusion.Diagramming.Layout Namespace

LaneGrid.SetColumnCount Method

ISwimlaneGrid.SetColumnCount implementation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void SetColumnCount (
    int value
)
```

Visual Basic Copy Code

```
Public Sub SetColumnCount( _
    value As Integer _
)
```

Parameters

value

See Also

LaneGrid Members
LaneGrid Class
MindFusion.Diagramming.Layout Namespace

LaneGrid.SetColumnWidth Method

ISwimlaneGrid.SetColumnCount implementation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void SetColumnWidth (
    int index,
    double value
)
```

Visual Basic Copy Code

```
Public Sub SetColumnWidth( _
    index As Integer, _
    value As Double _
)
```

Parameters

index

value

See Also

LaneGrid Members
LaneGrid Class
MindFusion.Diagramming.Layout Namespace

LaneGrid.SetLeftMargin Method

ISwimlaneGrid.SetLeftMargin implementation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void SetLeftMargin (
    double value
)
```

Visual Basic Copy Code

```
Public Sub SetLeftMargin( _
    value As Double _
)
```

Parameters

value

See Also

LaneGrid.SetRowCount Method

ISwimlaneGrid.SetRowCount implementation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void SetRowCount (
    int value
)
```

Visual Basic Copy Code

```
Public Sub SetRowCount( _
    value As Integer _
)
```

Parameters

value

See Also

LaneGrid Members
LaneGrid Class
MindFusion.Diagramming.Layout Namespace

LaneGrid.SetRowHeight Method

ISwimlaneGrid.SetRowHeight implementation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void SetRowHeight (
    int index,
    double value
)
```

Visual Basic Copy Code

```
Public Sub SetRowHeight( _
    index As Integer, _
    value As Double _
)
```

Parameters

index

value

See Also

LaneGrid Members
LaneGrid Class

LaneGrid.SetTopMargin Method

ISwimlaneGrid.SetTopMargin implementation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void SetTopMargin (
    double value
)
```

Visual Basic Copy Code

```
Public Sub SetTopMargin( _
    value As Double _
)
```

Parameters

value

See Also

LaneGrid Members
LaneGrid Class
MindFusion.Diagramming.Layout Namespace

LaneType Enumeration

Specifies where **FlowLayout** gets the swimlane dimensions from.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum LaneType
```

Visual Basic Copy Code

```
Public Enum LaneType
```

Members

	Member name	Description
	LaneGrid	**FlowLayout** gets the swimlane dimensions from Diagram.LaneGrid.
	LaneNodes	**FlowLayout** interprets locked background nodes as swimlanes.
	Custom	**FlowLayout** should get the swimlane dimensions from the array of rectangles passed as an argument to Arrange.

See Also

MindFusion.Diagramming.Layout Namespace

LayeredLayout Class

Implements a layered graph layout algorithm.

Syntax

C# Copy Code

```
public class LayeredLayout : Layout
```

Visual Basic Copy Code

```
Public Class LayeredLayout
    Inherits Layout
```

Remarks

This algorithm assigns diagram nodes to distinct horizontal or vertical layers. While arranging the layers, the layout routine seeks to meet the following criteria:

- connected nodes must be placed close together;
- links must flow in one direction if possible;
- links must cross as few layers as possible;
- links must not cross other links;

To apply this layout to a diagram, create a **LayeredLayout** instance, set its members and invoke the Arrange method. Arrange lays out all subgraphs in a diagram, unless the Root property is set.

The final appearance of arranged graphs depends mostly on the Orientation and Direction properties which define the overall orientation of the layout. The LayerDistance and NodeDistance properties specify how much space to leave between adjacent layers and adjacent nodes in a layer.

The performance of **LayeredLayout** is dependent on the number of links in the diagram. At its first stage, the algorithm looks for the longest sequence of links in the graph and builds the node layers along it. That stage can be the most time consuming if there are a lot of links in a graph, so there is the TimeLimit property provided to control how much time the layout routine is allowed to spend on path-finding. Next, if there are layers much longer than the other ones and SplitLayers is enabled, nodes from these layers are moved to new ones so the graph width is uniformly distributed.

The layout routine continues by shuffling nodes and link points across their layers, so the criteria mentioned above are approximated. Finally, link ends are aligned to anchor points if Anchoring is enabled and the distance between outermost links is decreased as set in LinksCompactFactor.

The layered graph layout algorithm produces best results if it can place link ends at arbitrary positions over nodes' outlines, that is, when using the Ignore value. If your application logic requires links to be always attached to anchor points, use the other two values. In such cases **Reassign** would yield better results.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.LayeredLayout

See Also

LayeredLayout Members
MindFusion.Diagramming.Layout Namespace

LayeredLayout Members

Implements a layered graph layout algorithm.

The following tables list the members exposed by the LayeredLayout type.

Public Constructors

	Name	Description
	LayeredLayout	Overloaded. Initializes a new instance of the **LayeredLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	Direction	Gets or sets the direction of the arranged diagram.
	EnableCustomLayers	Gets or sets a value indicating whether to indicating whether the layered layout will honor the LayeredLayoutTraits.Layer flags.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	EndNode	Gets or sets the node that should be placed in the last layer of the arranged graph.
	EnforceLinkFlow	Gets or sets a value indicating whether to attempt to force all links to point to the same direction. If the graph being arranged contains cycles, some of the links will still point backwards.
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	IgnoreNodeSize	Gets or sets a value indicating whether the distance between adjacent nodes in the final layout should be independent of the nodes' size.
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	KeepLinkShapes	Gets or sets whether to preserve initial Shape value of links when LinkType is set to Cascading.
	KeepRootPosition	Gets or sets a value indicating whether to preserve the position of the root node.
	LayerDistance	Gets or sets the desired distance between layer axis lines.
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	LinkPadding	Gets or sets the distance between parallel cascading links.
	LinksCompactFactor	Gets or sets a value indicating how much to compact the outermost links.
	LinkType	Gets or sets the appearance of the links in the final layout.
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	NodeDistance	Gets or sets the desired distance between adjacent nodes in a layer.
	Orientation	Gets or sets the overall orientation of the arranged diagram.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)

	Root	Gets or sets a node indicating which subgraph of the diagram must be laid out.
	SplitLayers	Gets or sets a value indicating whether to split layers that are much wider than the other layers.
	StableSort	Gets or sets a value indicating whether to apply stable sorting during the layout algorithm.
	StartNode	Gets or sets the node that should be placed in the first layer of the arranged graph.
	Statistics	Gets layout-specific information about the nodes and links participating in the layout.
	StraightenLongLinks	Gets or sets whether the layout algorithm should strive to straighten links that cross more than one layer of the graph.
	SwapPairsIterations	Gets or sets how many times to perform pair-swapping iterations on each layer.
	TimeLimit	Gets or sets a time-out value for the path-finding part of the layered layout algorithm.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout Constructor

Initializes a new instance of the LayeredLayout class.

Overload List

Name	Description
LayeredLayout ()	Initializes a new instance of the LayeredLayout class with the default settings.
LayeredLayout (Orientation, Double, Double, Size)	Initializes a new instance of the LayeredLayout class with the specified settings.

See Also

LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout Constructor ()

Initializes a new instance of the LayeredLayout class with the default settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public LayeredLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

LayeredLayout Constructor Overload List
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout Constructor (Orientation, Double, Double, Size)

Initializes a new instance of the LayeredLayout class with the specified settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public LayeredLayout (
    Orientation orientation,
    double layerDistance,
    double nodeDistance,
    Size margins
)
```

Visual Basic Copy Code

```
Public New ( _
    orientation As Orientation, _
    layerDistance As Double, _
    nodeDistance As Double, _
    margins As Size _
)
```

Parameters

orientation
The orientation of the layout.

layerDistance
The distance between adjacent layers.

nodeDistance
The distance between adjacent nodes within the same layer.

margins
The margin between the produced graph and the left side of the diagram bounding rectangle.

See Also

LayeredLayout Constructor Overload List
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout Properties

LayeredLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.Direction Property

Gets or sets the direction of the arranged diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Direction Direction { get; set; }
```

Visual Basic Copy Code

```
Public Property Direction As Direction
```

Property Value

A member of the Direction enumeration. The default is **Straight**.

Remarks

If Orientation is set to Horizontal, Direction.Straight stands for "left to right" and **Direction.Reversed** for "right to left". If **Orientation** is set to **Vertical**, **Direction.Straight** stands for "top to bottom" and **Direction.Reversed** for "bottom to top".

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.EnableCustomLayers Property

Gets or sets a value indicating whether to indicating whether the layered layout will honor the LayeredLayoutTraits.Layer flags.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool EnableCustomLayers { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableCustomLayers As Boolean
```

Property Value

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.EndNode Property

Gets or sets the node that should be placed in the last layer of the arranged graph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public DiagramNode EndNode { get; set; }
```

Visual Basic Copy Code

```
Public Property EndNode As DiagramNode
```

Property Value

A DiagramNode object representing the end node.

Remarks

This value is honored only if StartNode is specified too.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.EnforceLinkFlow Property

Gets or sets a value indicating whether to attempt to force all links to point to the same direction. If the graph being arranged contains cycles, some of the links will still point backwards.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool EnforceLinkFlow { get; set; }
```

Visual Basic Copy Code

```
Public Property EnforceLinkFlow As Boolean
```

Property Value

Remarks

EnableCustomLayers and SplitLayers are ignored if this property is set to **true**.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace
StraightenLongLinks Property
OneWayLayout Class

LayeredLayout.IgnoreNodeSize Property

Gets or sets a value indicating whether the distance between adjacent nodes in the final layout should be independent of the nodes' size.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool IgnoreNodeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property IgnoreNodeSize As Boolean
```

Property Value

true to ignore the node size; otherwise, **false**. The default is **true**.

Remarks

At this time, considering the size of nodes deteriorates the performance of the algorithm, so that's disabled by default.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.KeepLinkShapes Property

Gets or sets whether to preserve initial Shape value of links when LinkType is set to Cascading.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool KeepLinkShapes { get; set; }
```

Visual Basic Copy Code

```
Public Property KeepLinkShapes As Boolean
```

Property Value

true to preserve link Shape values, or **false** otherwise.

Remarks

If this property is enabled, the orthogonal path applied by Cascading LinkType is approximated using respective link shape.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.KeepRootPosition Property

Gets or sets a value indicating whether to preserve the position of the root node.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool KeepRootPosition { get; set; }
```

Visual Basic Copy Code

```
Public Property KeepRootPosition As Boolean
```

Property Value

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.LayerDistance Property

Gets or sets the desired distance between layer axis lines.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double LayerDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property LayerDistance As Double
```

Property Value

A **double** value specifying the distance between layers. The default is 25.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.LinkPadding Property

Gets or sets the distance between parallel cascading links.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double LinkPadding { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkPadding As Double
```

Property Value

A **double** value specifying the padding space between cascading links. The default value is 0.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.LinksCompactFactor Property

Gets or sets a value indicating how much to compact the outermost links.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double LinksCompactFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property LinksCompactFactor As Double
```

Property Value

A **double** number in the interval [0, 1]. The default is 1.

Remarks

Links and nodes on the same graph layer are distributed uniformly as specified via the NodeDistance value. However, if there are more than 2-3 outermost links on a layer, they usually take up too much space without improving the visual quality of the diagram layout. This property allows compacting such links to save space. The factor specifies roughly a fraction of **NodeDistance** to leave between outer links.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.LinkType Property

Gets or sets the appearance of the links in the final layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public LayeredLayoutLinkType LinkType { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkType As LayeredLayoutLinkType
```

Property Value

A member of the LayeredLayoutLinkType enumeration. The default is **Straight**.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace
StraightenLongLinks Property

LayeredLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Layered layout".

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.NodeDistance Property

Gets or sets the desired distance between adjacent nodes in a layer.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double NodeDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeDistance As Double
```

Property Value

A **double** value specifying the distance between adjacent nodes. The default is 20.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.Orientation Property

Gets or sets the overall orientation of the arranged diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic Copy Code

```
Public Property Orientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default is **Vertical**.

Remarks

Specify Orientation.Vertical to have links flow in vertical direction and **Orientation.Horizontal** to have links flow in horizontal direction. The orientation of the resulting layers is opposite to that of the links. Thus, if the link tends to go in a vertical direction, the nodes are distributed in horizontal layers.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace
Direction Property

LayeredLayout.Root Property

Gets or sets a node indicating which subgraph of the diagram must be laid out.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public DiagramNode Root { get; set; }
```

Visual Basic Copy Code

```
Public Property Root As DiagramNode
```

Property Value

An instance of the DiagramNode class. The default is a **null** reference (**Nothing** in Visual Basic).

Remarks

All links and nodes that can be reached from the root by following recursively its links form a graph. If the root is specified, only the graph containing it is laid out by Arrange.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.SplitLayers Property

Gets or sets a value indicating whether to split layers that are much wider than the other layers.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool SplitLayers { get; set; }
```

Visual Basic Copy Code

```
Public Property SplitLayers As Boolean
```

Property Value

true to split wide layers; otherwise, **false**. The default is **false**.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.StableSort Property

Gets or sets a value indicating whether to apply stable sorting during the layout algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool StableSort { get; set; }
```

Visual Basic Copy Code

```
Public Property StableSort As Boolean
```

Property Value

true to apply stable sorting algorithm, otherwise **false**. The default value is **false**.

Remarks

By default .NET sorts arrays using unstable sort, where if two elements have equal values, their order might not be the same between different runs of the sort method. This could lead to LayeredLayout not preserving the order of some nodes in a layer between different runs if there are more than one candidates for the same position. If you need the same order to be reproduced consistently, set **StableSort** to **true**.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.StartNode Property

Gets or sets the node that should be placed in the first layer of the arranged graph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public DiagramNode StartNode { get; set; }
```

Visual Basic Copy Code

```
Public Property StartNode As DiagramNode
```

Property Value

A DiagramNode object representing the start node.

Remarks

If there is no start node specified, the algorithm automatically selects one based on other LayeredLayout properties, such as EnforceLinkFlow.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace
EndNode Property

LayeredLayout.Statistics Property

Gets layout-specific information about the nodes and links participating in the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public LayeredLayoutStatistics Statistics { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Statistics As LayeredLayoutStatistics
```

Property Value

An instance of the LayeredLayoutStatistics class containing detailed information about the performed layout, such as the layers individual nodes were assigned to.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.StraightenLongLinks Property

Gets or sets whether the layout algorithm should strive to straighten links that cross more than one layer of the graph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool StraightenLongLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property StraightenLongLinks As Boolean
```

Property Value

true to straighten long links; otherwise **false**. The default value is **false**.

Remarks

If this property is enabled, links that cross multiple layers have at most two bends. Because of the additional horizontal alignment constraints, the arranged graph usually occupies a larger area than when not using **StraightenLongLinks**.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace
LinkType Property
EnforceLinkFlow Property

LayeredLayout.SwapPairsIterations Property

Gets or sets how many times to perform pair-swapping iterations on each layer.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int SwapPairsIterations { get; set; }
```

Visual Basic Copy Code

```
Public Property SwapPairsIterations As Integer
```

Property Value

An integer value specifying how many pair-swapping iterations to perform. The default is 5.

Remarks

The algorithm swaps node pairs on each layer aiming to decrease the number of crossing links. Setting a higher value to this property helps to decrease the number of crossings.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout.TimeLimit Property

Gets or sets a time-out value for the path-finding part of the layered layout algorithm.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public long TimeLimit { get; set; }
```

Visual Basic Copy Code

```
Public Property TimeLimit As Long
```

Property Value

An integer value specifying time-out in milliseconds. The default is 10000 (10 seconds).

Remarks

The layered layout algorithm arranges graph nodes and links along the longest path in the graph. Large graphs with many links could contain hundreds of thousands of possible paths, so finding the longest one would be very time-consuming. When the time-out set with this property occurs, the path-finding routine stops and the layout algorithm uses the longest path known at that moment. Usually, for graphs containing several hundreds of nodes, 1-2 seconds are enough to find a long-enough path to produce nice layered layout.

See Also

LayeredLayout Members
LayeredLayout Class
MindFusion.Diagramming.Layout Namespace

LayeredLayout Events

LayeredLayoutLinkType Enumeration

Specifies the style of the links after they are laid out.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum LayeredLayoutLinkType
```

Visual Basic Copy Code

```
Public Enum LayeredLayoutLinkType
```

Members

	Member name	Description
	Straight	Indicates straight links.
	Cascading	Indicates cascading links.

See Also

MindFusion.Diagramming.Layout Namespace

LayeredLayoutStatistics Class

Encapsulates information about a layered layout which has already complete.

Syntax

C# Copy Code

```
public sealed class LayeredLayoutStatistics
```

Visual Basic Copy Code

```
Public NotInheritable Class LayeredLayoutStatistics
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Layout.LayeredLayoutStatistics

See Also

LayeredLayoutStatistics Members
MindFusion.Diagramming.Layout Namespace

LayeredLayoutStatistics Members

Encapsulates information about a layered layout which has already complete.

The following tables list the members exposed by the LayeredLayoutStatistics type.

Public Properties

	Name	Description
	LayerCount	Gets the total number of layers in the layout. If more than one subgraph is laid out, this property returns the number of layers in the last subgraph.
	LinkLayerCount	Gets a mapping between links and the number of layers they run through.
	NodeGridPositions	Gets a mapping between nodes and their assigned grid positions.
	NodeLayerIndices	Gets a mapping between nodes and their assigned layers.

See Also

LayeredLayoutStatistics Class
MindFusion.Diagramming.Layout Namespace

LayeredLayoutStatistics Properties

LayeredLayoutStatistics.LayerCount Property

Gets the total number of layers in the layout. If more than one subgraph is laid out, this property returns the number of layers in the last subgraph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int LayerCount { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property LayerCount As Integer
```

Property Value

See Also

LayeredLayoutStatistics Members
LayeredLayoutStatistics Class
MindFusion.Diagramming.Layout Namespace

LayeredLayoutStatistics.LinkLayerCount Property

Gets a mapping between links and the number of layers they run through.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Dictionary<DiagramLink, Int32> LinkLayerCount { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property LinkLayerCount As Dictionary(Of DiagramLink, Int32)
```

Property Value

See Also

LayeredLayoutStatistics Members
LayeredLayoutStatistics Class
MindFusion.Diagramming.Layout Namespace

LayeredLayoutStatistics.NodeGridPositions Property

Gets a mapping between nodes and their assigned grid positions.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Dictionary<DiagramNode, Double> NodeGridPositions { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NodeGridPositions As Dictionary(Of DiagramNode, Double)
```

Property Value

See Also

LayeredLayoutStatistics Members
LayeredLayoutStatistics Class
MindFusion.Diagramming.Layout Namespace

LayeredLayoutStatistics.NodeLayerIndices Property

Gets a mapping between nodes and their assigned layers.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Dictionary<DiagramNode, Int32> NodeLayerIndices { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property NodeLayerIndices As Dictionary(Of DiagramNode, Int32)
```

Property Value

See Also

LayeredLayoutStatistics Members
LayeredLayoutStatistics Class
MindFusion.Diagramming.Layout Namespace

Layout Class

The base class for all layout classes.

Syntax

C# Copy Code

```
public abstract class Layout : Component, ILayout
```

Visual Basic Copy Code

```
Public MustInherit Class Layout
    Inherits Component
    Implements ILayout
```

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.AnnealLayout

MindFusion.Diagramming.Layout.CascadeLayout
MindFusion.Diagramming.Layout.CircularLayout
MindFusion.Diagramming.Layout.CompositeLayout
MindFusion.Diagramming.Layout.DecisionLayout
MindFusion.Diagramming.Layout.FlowchartLayout
MindFusion.Diagramming.Layout.FractalLayout
MindFusion.Diagramming.Layout.GridLayout
MindFusion.Diagramming.Layout.HierarchicalLayout
MindFusion.Diagramming.Layout.LayeredLayout
MindFusion.Diagramming.Layout.OneWayLayout
MindFusion.Diagramming.Layout.OrthogonalLayout
MindFusion.Diagramming.Layout.OrthogonalRouter
MindFusion.Diagramming.Layout.SpringLayout
MindFusion.Diagramming.Layout.SwimlaneLayout
MindFusion.Diagramming.Layout.TopologicalLayout
MindFusion.Diagramming.Layout.TreeLayout
MindFusion.Diagramming.Layout.TreeMapLayout
MindFusion.Diagramming.Layout.TriangularLayout

See Also

Layout Members
MindFusion.Diagramming.Layout Namespace

Layout Members

The base class for all layout classes.

The following tables list the members exposed by the Layout type.

Protected Constructors

	Name	Description
	Layout	Initializes a new instance of the **Layout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads.
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content.
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph.
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout.
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout.
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds.
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism.
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.
	Name	Gets a user-friendly name of the layout.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress.

	RepeatingLinksPadding	Specifies the padding distance between repeating links.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value for the Margins property, which may vary across different layouts.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance.

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance.
	ArrangeAnimated	Overloaded.
	ArrangeAsync	Overloaded.

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event.

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph.

See Also

Layout Class
MindFusion.Diagramming.Layout Namespace

Layout Constructor

Initializes a new instance of the Layout class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected Layout ()
```

Visual Basic Copy Code

```
Protected Sub New ()
```

See Also

Layout Class
MindFusion.Diagramming.Layout Namespace

Layout Methods

Layout.Arrange Method

Applies the layout to a Diagram instance.

Overload List

Name	Description
Layout.Arrange (Diagram)	Applies the layout to the specified **Diagram** instance.
Layout.Arrange (Diagram, DiagramItemCollection)	Applies the layout on the specified subset of objects from the specified **Diagram** instance.
Layout.Arrange (Diagram, DiagramItemCollection, Double, Double)	

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.Arrange Method (Diagram)

Applies the layout to the specified Diagram instance.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public virtual bool Arrange (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public Overridable Function Arrange( _
    diagram As Diagram _
) As Boolean
```

Parameters

diagram
The Diagram that must be arranged.

Return Value

true if the diagram has been arranged successfully; otherwise, **false**.

See Also

Arrange Method Overload List
Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.Arrange Method (Diagram, DiagramItemCollection)

Applies the layout on the specified subset of objects from the specified Diagram instance.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public virtual bool Arrange (
    Diagram diagram,
    DiagramItemCollection items
)
```

Visual Basic Copy Code

```
Public Overridable Function Arrange( _
    diagram As Diagram, _
    items As DiagramItemCollection _
) As Boolean
```

Parameters

diagram

The Diagram that must be arranged.

items

A collection of nodes and links from diagram that define the subgraph to arrange.

Return Value

true if the diagram has been arranged successfully; otherwise, **false**.

See Also

Arrange Method Overload List
Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.Arrange Method (Diagram, DiagramItemCollection, Double, Double)

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public virtual bool Arrange (
    Diagram diagram,
    DiagramItemCollection items,
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Public Overridable Function Arrange( _
    diagram As Diagram, _
    items As DiagramItemCollection, _
    dx As Double, _
    dy As Double _
) As Boolean
```

Parameters

diagram

items

dx

dy

Return Value

See Also

Arrange Method Overload List
Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.ArrangeAnimated Method

Overload List

Name	Description
Layout.ArrangeAnimated (Diagram, DiagramItemCollection, Int32, AnimationType, EasingType, Double, Double)	
Layout.ArrangeAnimated (Diagram, Int32, AnimationType, EasingType)	Applies the layout to the specified subset of items from the specified Diagram instance and animates items moving to their new positions.

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.ArrangeAnimated Method (Diagram, DiagramItemCollection, Int32, AnimationType, EasingType, Double, Double)

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public virtual bool ArrangeAnimated (
    Diagram diagram,
    DiagramItemCollection items,
    int duration,
    AnimationType animationType,
    EasingType easingType,
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Public Overridable Function ArrangeAnimated( _
    diagram As Diagram, _
    items As DiagramItemCollection, _
    duration As Integer, _
    animationType As AnimationType, _
    easingType As EasingType, _
    dx As Double, _
    dy As Double _
) As Boolean
```

Parameters

diagram

items

duration

animationType

easingType

dx

dy

Return Value

See Also

ArrangeAnimated Method Overload List
Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.ArrangeAnimated Method (Diagram, Int32, AnimationType, EasingType)

Applies the layout to the specified subset of items from the specified Diagram instance and animates items moving to their new positions.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public virtual bool ArrangeAnimated (
    Diagram diagram,
    int duration,
    AnimationType animationType,
    EasingType easingType
)
```

Visual Basic Copy Code

```
Public Overridable Function ArrangeAnimated( _
    diagram As Diagram, _
    duration As Integer, _
    animationType As AnimationType, _
    easingType As EasingType _
) As Boolean
```

Parameters

diagram
The Diagram that should be arranged.

duration
An integer, specifying the duration of the animation in milliseconds.

animationType
A member of the AnimationType enumeration, specifying the animation type to use.

easingType
A member of the EasingType enumeration, specifying the type of easing function to apply to the animation.

Return Value

true if the diagram has been arranged successfully; otherwise, false.

See Also

Layout.ArrangeAsync Method

Overload List

Name	Description
Layout.ArrangeAsync (Diagram)	Applies the layout asynchronously to the specified Diagram instance.
Layout.ArrangeAsync (Diagram, DiagramItemCollection)	Applies the layout asynchronously to the specified subset of items from the specified **Diagram** instance.
Layout.ArrangeAsync (Diagram, DiagramItemCollection, Double, Double)	

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.ArrangeAsync Method (Diagram)

Applies the layout asynchronously to the specified Diagram instance.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public async Task<Boolean> ArrangeAsync (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public Async Function ArrangeAsync( _
    diagram As Diagram _
) As Task(Of Boolean)
```

Parameters

diagram
The Diagram that should be arranged.

Return Value

true if the diagram has been arranged successfully, or **false** otherwise.

Remarks

This method is available only in the .NET 4.5 build of the *MindFusion.Diagramming.dll* assembly.

See Also

ArrangeAsync Method Overload List
Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.ArrangeAsync Method (Diagram, DiagramItemCollection)

Applies the layout asynchronously to the specified subset of items from the specified Diagram instance.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public async Task<Boolean> ArrangeAsync (
    Diagram diagram,
    DiagramItemCollection items
)
```

Visual Basic Copy Code

```
Public Async Function ArrangeAsync( _
    diagram As Diagram, _
    items As DiagramItemCollection _
) As Task(Of Boolean)
```

Parameters

diagram
The Diagram that should be arranged.

items
A collection of nodes and links from diagram that define the subgraph to arrange.

Return Value

true if the diagram has been arranged successfully, or **false** otherwise.

Remarks

This method is available only in the .NET 4.5 build of the *MindFusion.Diagramming.dll* assembly.

See Also

ArrangeAsync Method Overload List
Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.ArrangeAsync Method (Diagram, DiagramItemCollection, Double, Double)

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public virtual Task<Boolean> ArrangeAsync (
    Diagram diagram,
    DiagramItemCollection items,
    double dx = 0,
    double dy = 0
)
```

Visual Basic Copy Code

```
Public Overridable Function ArrangeAsync( _
    diagram As Diagram, _
    items As DiagramItemCollection, _
    Optional dx As Double = 0, _
```

```
    Optional dy As Double = 0 _
) As Task(Of Boolean)
```

Parameters

diagram

items

dx
Optional.

dy
Optional.

Return Value

See Also

ArrangeAsync Method Overload List
Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.OnLayoutComplete Method

Raises the LayoutComplete event.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected void OnLayoutComplete (
    EventArgs e
)
```

Visual Basic Copy Code

```
Protected Sub OnLayoutComplete( _
    e As EventArgs _
)
```

Parameters

e
An object providing the event data.

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout Properties

Layout.Anchoring Property

Gets or sets a value indicating how to align links to the anchor points of graph nodes.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Anchoring Anchoring { get; set; }
```

Visual Basic Copy Code

```
Public Property Anchoring As Anchoring
```

Property Value

A member of the Anchoring enumeration. The default is **Ignore**.

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.DefaultMarginValue Property

Gets the default value for the Margins property, which may vary across different layouts.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected abstract Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected MustOverride ReadOnly Property DefaultMarginValue As Size
```

Property Value

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.EnableParallelism Property

Gets or sets a value indicating whether to arrange subgraphs on separate threads.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public virtual bool EnableParallelism { get; set; }
```

Visual Basic Copy Code

```
Public Overridable Property EnableParallelism As Boolean
```

Property Value

true to enable parallelism; otherwise, **false**.

Remarks

Available only in .NET 4.0 versions and later.

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.GrowToFit Property

Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool GrowToFit { get; set; }
```

Visual Basic Copy Code

```
Public Property GrowToFit As Boolean
```

Property Value

true to grow diagram's Bounds if the arranged items would not fit, or **false** otherwise. The default value is **true**.

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.KeepGroupLayout Property

Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool KeepGroupLayout { get; set; }
```

Visual Basic Copy Code

```
Public Property KeepGroupLayout As Boolean
```

Property Value

true if each Group should be processed as an integral graph node; otherwise, **false**. The default is **false**.

Remarks

Enabling this property has an effect similar to setting the IgnoreLayout property of group's subordinated nodes to **true**. The two methods differ in how the size of the graph node corresponding to a group is calculated. If **KeepGroupLayout** is enabled, the union of all bounding rectangles of grouped items is used to determine the size of the graph node. When the **IgnoreLayout** property of subordinated items is enabled, the bounding rectangle of the main node defines the size of the graph node corresponding to the group.

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.LayoutLink Property

Gets or sets a delegate, which is invoked when the position of a link is changed by the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public virtual LayoutLink LayoutLink { get; set; }
```

Visual Basic Copy Code

```
Public Overridable Property LayoutLink As LayoutLink
```

Property Value

An instance of the LayoutLink delegate.

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.LayoutNode Property

Gets or sets a delegate, which is invoked when the position of a node is changed by the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public virtual LayoutNode LayoutNode { get; set; }
```

Visual Basic Copy Code

```
Public Overridable Property LayoutNode As LayoutNode
```

Property Value

An instance of the **LayoutNode** delegate.

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.Margins Property

Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Size Margins { get; set; }
```

Visual Basic

```
Public Property Margins As Size
```

Property Value

Remarks

This value is also set as the distance between unconnected sub-graphs when they are placed as specified via MultipleGraphsPlacement.

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.MaxDegreeOfParallelism Property

Gets or sets the maximum degree of parallelism.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public virtual int MaxDegreeOfParallelism { get; set; }
```

Visual Basic

```
Public Overridable Property MaxDegreeOfParallelism As Integer
```

Property Value

An integer value specifying the degree.

Remarks

Applicable only if EnableParallelization is set to true. Available only in .NET 4.0 versions and later.

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.MultipleGraphsPlacement Property

Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public MultipleGraphsPlacement MultipleGraphsPlacement { get; set; }
```

Visual Basic

```
Public Property MultipleGraphsPlacement As MultipleGraphsPlacement
```

Property Value

A member of the MultipleGraphsPlacement enumeration. The default value is **Horizontal**.

Remarks

If the diagram contains several unconnected subgraphs, each of them is processed by the algorithm independently from the others. This property specifies how such subgraphs should be placed one relatively to another. The possible options are to arrange them in a row, in a column, or in such a way that the diagram covers a minimal area.

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public abstract string Name { get; }
```

Visual Basic Copy Code

```
Public MustOverride ReadOnly Property Name As String
```

Property Value

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.Progress Property

Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public virtual LayoutProgress Progress { get; set; }
```

Visual Basic Copy Code

```
Public Overridable Property Progress As LayoutProgress
```

Property Value

A LayoutProgress delegate method to call while the algorithm proceeds.

Remarks

It might take a lot of time to arrange large diagrams or run more iterations of the algorithm to get better results. In such situations an application can be notified about the progress of the layout routine by assigning a delegate method to this property. On its turn the application could update the UI to indicate the current progress to the users.

Example

The following example updates a label text to indicate the progress of a layout algorithm. The *CreateLayout* method instantiates a layout object from a concrete class.

C# Copy Code

```
Layout layout = CreateLayout();
layout.Progress = new LayoutProgress(onProgress);
layout.Arrange(diagram);

// ...

void onProgress(int c, int n)
{
    label1.Text = (100*c/n).ToString() + "% complete";
    label1.Refresh();
}
```

Visual Basic Copy Code

```
Dim layout As Layout = CreateLayout()
layout.Progress = New LayoutProgress(AddressOf onProgress)
layout.Arrange(diagram)

' ...

Sub onProgress(ByVal c As Integer, ByVal n As Integer)

    label1.Text = (100 * c / n).ToString() + "% complete"
    label1.Refresh()

End Sub
```

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.RepeatingLinksPadding Property

Specifies the padding distance between repeating links.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double RepeatingLinksPadding { get; set; }
```

Visual Basic Copy Code

```
Public Property RepeatingLinksPadding As Double
```

Property Value

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout.SuppressCompleteEvent Property

Specifies whether raising the LayoutComplete event should be suppressed for this layout instance.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected bool SuppressCompleteEvent { get; set; }
```

Visual Basic Copy Code

```
Protected Property SuppressCompleteEvent As Boolean
```

Property Value

See Also

Layout Members
Layout Class
MindFusion.Diagramming.Layout Namespace

Layout Events

LayoutLink Delegate

Represents methods that are called by the layout algorithms when the position of a link changes.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public delegate void LayoutLink (
    DiagramLink link
)
```

Visual Basic Copy Code

```
Public Delegate Sub LayoutLink( _
    link As DiagramLink _
)
```

Parameters

link
A reference to the link being laid out.

See Also

MindFusion.Diagramming.Layout Namespace

LayoutNode Delegate

Represents methods that are called by the layout algorithms when the position of a node changes.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public delegate void LayoutNode (
    DiagramNode node,
    Rect oldBounds
)
```

Visual Basic Copy Code

```
Public Delegate Sub LayoutNode( _
    node As DiagramNode, _
    oldBounds As Rect _
)
```

Parameters

node
A reference to the node being laid out.

oldBounds
The bounding rectangle of the node before it was repositioned by the layouting algorithm.

See Also

MindFusion.Diagramming.Layout Namespace

LayoutProgress Delegate

Represents methods that are called to report on progress of layout algorithms.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public delegate void LayoutProgress (
    int current,
    int total
)
```

Visual Basic Copy Code

```
Public Delegate Sub LayoutProgress( _
    current As Integer, _
    total As Integer _
)
```

Parameters

current
An integer value specifying the current relative progress of the layout algorithm.

total
An integer value specifying the total number of steps required by the layouting algorithm in order to finish.

See Also

MindFusion.Diagramming.Layout Namespace

LayoutSubgraph Delegate

Represents methods that are called by the layout algorithms in order to perform custom arrangement of a subgraph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public delegate bool LayoutSubgraph (
    Diagram diagram,
    DiagramItemCollection items
)
```

Visual Basic Copy Code

```
Public Delegate Function LayoutSubgraph( _
    diagram As Diagram, _
    items As DiagramItemCollection _
) As Boolean
```

Parameters

diagram
The diagram being arranged.

items
The subset of items to arrange.

Return Value

true if the items were arranged; otherwise, **false**.

See Also

MindFusion.Diagramming.Layout Namespace

MultipleGraphsPlacement Enumeration

Specifies graph orientation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum MultipleGraphsPlacement
```

Visual Basic Copy Code

```
Public Enum MultipleGraphsPlacement
```

Members

	Member name	Description
	Vertical	Indicates that graphs should be placed in a column.
	Horizontal	Indicates that graphs should be placed in a row.
	MinimalArea	Indicates that graphs should be placed so that they cover minimal area.
	VerticalSortDescending	Indicates that graphs should be placed in a column and sorted by the number of vertices they contain in a descending order.
	VerticalSortAscending	Indicates that graphs should be placed in a column and sorted by the number of vertices they contain in an ascending order.
	HorizontalSortDescending	Indicates that graphs should be placed in a row and sorted by the number of vertices they contain in a descending order.

	HorizontalSortAscending	Indicates that graphs should be placed in a row and sorted by the number of vertices they contain in an ascending order.

See Also

MindFusion.Diagramming.Layout Namespace

OneWayLayout Class

Implements a one-way graph layout algorithm.

Syntax

C# Copy Code

```
public class OneWayLayout : Layout
```

Visual Basic Copy Code

```
Public Class OneWayLayout
    Inherits Layout
```

Remarks

The one-way layout ensures that links enter into nodes from the same general direction and exit them from the opposite side. If the graph contains cycles, some links bend around the nodes to keep the enter/exit direction consistent. The algorithm aims to minimize the number of such links.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.OneWayLayout

See Also

OneWayLayout Members
MindFusion.Diagramming.Layout Namespace

OneWayLayout Members

Implements a one-way graph layout algorithm.

The following tables list the members exposed by the OneWayLayout type.

Public Constructors

	Name	Description
	OneWayLayout	Initializes a new instance of the **OneWayLayout** class with the default settings.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)

	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	NodeDistance	Gets or sets the minimum spacing between adjacent nodes in the final layout.
	Orientation	Gets or sets the orientation of the arranged graph.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	RoundedLinks	Gets or sets a value indicating whether the layout should smooth link bends.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description

	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

OneWayLayout Class
MindFusion.Diagramming.Layout Namespace

OneWayLayout Constructor

Initializes a new instance of the OneWayLayout class with the default settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public OneWayLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

OneWayLayout Class
MindFusion.Diagramming.Layout Namespace

OneWayLayout Properties

OneWayLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

OneWayLayout Members
OneWayLayout Class
MindFusion.Diagramming.Layout Namespace

OneWayLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "One-way layout".

See Also

OneWayLayout Members
OneWayLayout Class
MindFusion.Diagramming.Layout Namespace

OneWayLayout.NodeDistance Property

Gets or sets the minimum spacing between adjacent nodes in the final layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double NodeDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeDistance As Double
```

Property Value

A float number specifying the distance. The default is 20.

See Also

OneWayLayout Members
OneWayLayout Class
MindFusion.Diagramming.Layout Namespace

OneWayLayout.Orientation Property

Gets or sets the orientation of the arranged graph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic Copy Code

```
Public Property Orientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default is **Vertical**.

See Also

OneWayLayout Members
OneWayLayout Class
MindFusion.Diagramming.Layout Namespace

OneWayLayout.RoundedLinks Property

Gets or sets a value indicating whether the layout should smooth link bends.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool RoundedLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property RoundedLinks As Boolean
```

Property Value

true to smooth link bends; otherwise, **false**.

See Also

OneWayLayout Members
OneWayLayout Class
MindFusion.Diagramming.Layout Namespace

OneWayLayout Events

Orientation Enumeration

Specifies axial orientation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum Orientation
```

Visual Basic Copy Code

```
Public Enum Orientation
```

Members

	Member name	Description
	Vertical	Indicates vertical orientation.

	Horizontal	Indicates horizontal orientation.

See Also

MindFusion.Diagramming.Layout Namespace

OrthogonalLayout Class

Implements an orthogonal graph layout algorithm. Each link is drawn as a chain of alternating horizontal and vertical segments. Nodes are placed in a way that facilitates few links bends and crossings.

Syntax

C# Copy Code

```
public class OrthogonalLayout : Layout
```

Visual Basic Copy Code

```
Public Class OrthogonalLayout
    Inherits Layout
```

Remarks

This algorithm was designed for planar graphs where the nodes have at most four incident links, and produces best results with such graphs as input. It can arrange any arbitrary graph as well, by adding some dummy nodes and links to planarize the graph and reduce its vertex degree, and removing them after the layout. However this might lead to using larger area than necessary, due to the space occupied by dummy elements.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.OrthogonalLayout

See Also

OrthogonalLayout Members
MindFusion.Diagramming.Layout Namespace

OrthogonalLayout Members

Implements an orthogonal graph layout algorithm. Each link is drawn as a chain of alternating horizontal and vertical segments. Nodes are placed in a way that facilitates few links bends and crossings.

The following tables list the members exposed by the OrthogonalLayout type.

Public Constructors

	Name	Description
	OrthogonalLayout	Initializes a new instance of the **OrthogonalLayout** class with the default settings.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	Directed	Gets or sets a value indicating whether the graph being laid out is directed.

	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MinimizeLinkBends	Gets or sets a value indicating whether the layout should minimize link bends.
	MinLaneSize	Gets or sets a value indicating the minimum size of a lane.
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	Padding	Gets or sets a value indicating how much space to leave between nodes in adjacent lanes.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	Refine	Gets or sets a value indicating whether to refine the layout.
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

OrthogonalLayout Class
MindFusion.Diagramming.Layout Namespace

OrthogonalLayout Constructor

Initializes a new instance of the OrthogonalLayout class with the default settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public OrthogonalLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

OrthogonalLayout Class
MindFusion.Diagramming.Layout Namespace

OrthogonalLayout Properties

OrthogonalLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

OrthogonalLayout Members
OrthogonalLayout Class
MindFusion.Diagramming.Layout Namespace

OrthogonalLayout.Directed Property

Gets or sets a value indicating whether the graph being laid out is directed.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool Directed { get; set; }
```

Visual Basic Copy Code

```
Public Property Directed As Boolean
```

Property Value

true to consider the direction of graph edges, **false** to ignore it; the default value is **false**.

See Also

OrthogonalLayout Members
OrthogonalLayout Class
MindFusion.Diagramming.Layout Namespace

OrthogonalLayout.MinimizeLinkBends Property

Gets or sets a value indicating whether the layout should minimize link bends.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool MinimizeLinkBends { get; set; }
```

Visual Basic Copy Code

```
Public Property MinimizeLinkBends As Boolean
```

Property Value

true to minimize link bends, otherwise **false**. The default value is **true**.

Remarks

If this option is disabled, the layout class applies a fast linear time algorithm, however the layouts produced usually contain more bending links and occupy more space. If enabled, the layout class applies an algorithm of polynomial complexity and runs much slower.

See Also

OrthogonalLayout Members
OrthogonalLayout Class
MindFusion.Diagramming.Layout Namespace

OrthogonalLayout.MinLaneSize Property

Gets or sets a value indicating the minimum size of a lane.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double MinLaneSize { get; set; }
```

Visual Basic Copy Code

```
Public Property MinLaneSize As Double
```

Property Value

A **double** value specifying the minimum lane size. The default value is 10.

Remarks

The result of the layout's Arrange is a grid that contains the diagram's nodes and links. This property specifies the size of lanes which does not contain nodes, but only links.

See Also

OrthogonalLayout Members
OrthogonalLayout Class
MindFusion.Diagramming.Layout Namespace

OrthogonalLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Orthogonal layout".

See Also

OrthogonalLayout Members
OrthogonalLayout Class
MindFusion.Diagramming.Layout Namespace

OrthogonalLayout.Padding Property

Gets or sets a value indicating how much space to leave between nodes in adjacent lanes.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double Padding { get; set; }
```

Visual Basic Copy Code

```
Public Property Padding As Double
```

Property Value

A **double** value specifying the padding. The default is 15.

See Also

OrthogonalLayout Members
OrthogonalLayout Class
MindFusion.Diagramming.Layout Namespace

OrthogonalLayout.Refine Property

Gets or sets a value indicating whether to refine the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool Refine { get; set; }
```

Visual Basic Copy Code

```
Public Property Refine As Boolean
```

Property Value

true to run a refinement pass, otherwise **false**. The default value is **true**.

See Also

OrthogonalLayout Members
OrthogonalLayout Class
MindFusion.Diagramming.Layout Namespace

OrthogonalLayout Events

OrthogonalRouter Class

Implements an orthogonal link routing algorithm.

Syntax

C# Copy Code

```
public class OrthogonalRouter : Layout
```

Visual Basic Copy Code

```
Public Class OrthogonalRouter
    Inherits Layout
```

Remarks

OrthogonalRouter is a secondary layout algorithm that can be used to arrange links after an initial node arrangement has already been applied. The orthogonal layout is useful when there are much more links than nodes in a graph. The algorithm strives to achieve the following criteria, while preserving as much of the initial node configuration as possible.

- links must not overlap;
- only vertical and horizontal routing lines are used;
- graph routing is performed with respect to the specified main layout direction;

- links crossings are minimized;
- bends are minimized;

To arrange a diagram, create an **OrthogonalRouter** instance, set its properties and call the Arrange method.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.OrthogonalRouter

See Also

OrthogonalRouter Members
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter Members

Implements an orthogonal link routing algorithm.

The following tables list the members exposed by the OrthogonalRouter type.

Public Constructors

	Name	Description
	OrthogonalRouter	Initializes a new instance of the **OrthogonalRouter** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	BendDistance	Gets or sets how far from its adjacent node a link should bend.
	Direction	Gets or sets the direction of the arranged diagram when BaseLayout is set to Layered.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	EndOrientation	Gets or sets the desired orientation of the last segments of routed links.
	GridSize	Gets or sets the distance between adjacent grid points when BaseLayout is set to Grid.
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayerDistance	Gets or sets the desired distance between layer axis lines.
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from

		Layout.)
	Name	Gets a user-friendly name of the layout.
	NodeDistance	Gets or sets the desired distance between adjacent nodes in a layer.
	Orientation	Gets or sets the overall orientation of the arranged diagram when BaseLayout is set to Layered.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	PutEndPointsAtNodeBorders	Gets or sets a value indicating whether the end points of a link should be placed at the outlines of the nodes connected by that link.
	RandomSeed	Gets or sets the randomization seed to use for generating random node positions.
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	StartOrientation	Gets or sets the desired orientation of the first segments of routed links.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter Constructor

Initializes a new instance of the OrthogonalRouter class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public OrthogonalRouter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter Properties

OrthogonalRouter.BendDistance Property

Gets or sets how far from its adjacent node a link should bend.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double BendDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property BendDistance As Double
```

Property Value

A **double** value specifying the distance from a link's first bend to its adjacent node. The default is 0, specifying that the algorithm should automatically choose a value for this parameter.

Remarks

This should be a value between 0 and 1, specifying a distance proportional to the node size.

See Also

OrthogonalRouter Members
OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(10, 10).

See Also

OrthogonalRouter Members
OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter.Direction Property

Gets or sets the direction of the arranged diagram when BaseLayout is set to Layered.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Direction Direction { get; set; }
```

Visual Basic Copy Code

```
Public Property Direction As Direction
```

Property Value

Remarks

This value is assigned to the Direction property of the base LayeredLayout.

See Also

OrthogonalRouter Members
OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter.EndOrientation Property

Gets or sets the desired orientation of the last segments of routed links.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Orientation EndOrientation { get; set; }
```

Visual Basic Copy Code

```
Public Property EndOrientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default value is **Auto**.

Remarks

If this property is set to Horizontal, links will end at the left or right sides of nodes. If set to **Vertical**, links will end at the top or bottom sides.

See Also

OrthogonalRouter Members
OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace
StartOrientation Property

OrthogonalRouter.GridSize Property

Gets or sets the distance between adjacent grid points when BaseLayout is set to Grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double GridSize { get; set; }
```

Visual Basic Copy Code

```
Public Property GridSize As Double
```

Property Value

Remarks

This value is assigned to the GridSize property of the base GridLayout.

See Also

OrthogonalRouter Members
OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter.LayerDistance Property

Gets or sets the desired distance between layer axis lines.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double LayerDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property LayerDistance As Double
```

Property Value

Remarks

Gets or sets the distance between layers when BaseLayout is set to Layered or Cascading.

See Also

OrthogonalRouter Members
OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

See Also

OrthogonalRouter Members
OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter.NodeDistance Property

Gets or sets the desired distance between adjacent nodes in a layer.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double NodeDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeDistance As Double
```

Property Value

Remarks

Gets or sets the distance between nodes in the same layer when BaseLayout is set to Layered or Cascading.

See Also

OrthogonalRouter Members
OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter.Orientation Property

Gets or sets the overall orientation of the arranged diagram when BaseLayout is set to Layered.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic — Copy Code

```
Public Property Orientation As Orientation
```

Property Value

Remarks

This value is assigned to the Orientation property of the base LayeredLayout.

See Also

OrthogonalRouter Members
OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter.PutEndPointsAtNodeBorders Property

Gets or sets a value indicating whether the end points of a link should be placed at the outlines of the nodes connected by that link.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# — Copy Code

```
public bool PutEndPointsAtNodeBorders { get; set; }
```

Visual Basic — Copy Code

```
Public Property PutEndPointsAtNodeBorders As Boolean
```

Property Value

See Also

OrthogonalRouter Members
OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace

OrthogonalRouter.RandomSeed Property

Gets or sets the randomization seed to use for generating random node positions.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# — Copy Code

```
public int RandomSeed { get; set; }
```

Visual Basic — Copy Code

```
Public Property RandomSeed As Integer
```

Property Value

See Also

OrthogonalRouter.StartOrientation Property

Gets or sets the desired orientation of the first segments of routed links.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Orientation StartOrientation { get; set; }
```

Visual Basic Copy Code

```
Public Property StartOrientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default value is **Auto**.

Remarks

If this property is set to Horizontal, links will end at the left or right sides of nodes. If set to **Vertical**, links will end at the top or bottom sides.

See Also

OrthogonalRouter Members
OrthogonalRouter Class
MindFusion.Diagramming.Layout Namespace
EndOrientation Property

OrthogonalRouter Events

SpringLayout Class

Implements the Spring-Embedder graph layout algorithm.

Syntax

C# Copy Code

```
public class SpringLayout : Layout
```

Visual Basic Copy Code

```
Public Class SpringLayout
    Inherits Layout
```

Remarks

Spring-Embedder produces layouts having uniform distribution of nodes by simulating a physical system in which nodes repulse each other and the links between them act as confining springs. Nodes are moved around in an iterative process. The forces that act on a node are calculated, taking in account the positions of surrounding nodes and links from the previous iteration.

To apply this layout to a diagram, call the Arrange method of a **SpringLayout** instance. Arrange lays out all subgraphs in a diagram, unless the Root property is set.

If the LayoutTraits value associated with a node through SpringLayoutTraits.Frozen is **true**, **SpringLayout** considers the repulsion forces exerted by the node on other nodes, but ignores forces acting on the node and keeps it in place.

The final appearance of arranged graphs depends mostly on the values of EnableClusters and NodeDistance properties that specify how nodes are distributed by the layout routine. Enabling the MinimizeCrossings property helps to decrease the number of link crossings, though that makes the algorithm run somewhat slower.

The layout routine runs as much iterations as set in IterationCount, calculating better node positions at each step. When that stage of the algorithm completes, if Anchoring is enabled, link ends are aligned to nodes' anchor points.

The Spring-Embedder layout algorithm produces best results if it can place link ends at arbitrary positions over nodes' outlines, that is, when using the Ignore value. If your application logic requires links to be always attached to anchor points, use the other two values. In such cases **Reassign** would yield better results.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.SpringLayout

See Also

SpringLayout Members
MindFusion.Diagramming.Layout Namespace

SpringLayout Members

Implements the Spring-Embedder graph layout algorithm.

The following tables list the members exposed by the SpringLayout type.

Public Constructors

	Name	Description
	SpringLayout	Overloaded. Initializes a new instance of the **SpringLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	EnableClusters	Gets or sets a value indicating whether to enable allocation of nodes to distinct clusters.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	IterationCount	Gets or sets for how many iterations to run the layout routine.
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	Layout3D	Specifies whether the layout should set the 3D coordinates of nodes for display inside **DiagramView3D**.
	LayoutArea	Gets or sets the size of the layout area.
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutMethod	Specifies what energy model to use for the simulation.
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	LinkType	Gets or sets the appearance of the links in the final layout.

	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MinimizeCrossings	Gets or sets a value indicating whether crossing minimization mode is enabled.
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	NodeDistance	Gets or sets the desired distance between nodes.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	Randomize	Gets or sets a value indicating whether nodes should be placed at random positions when the layout routine starts.
	RandomSeed	Gets or sets the randomization seed to use for generating random node positions.
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	RepulsionFactor	Gets or sets the multiplier applied to repulsion forces between nodes.
	Root	Gets or sets a node indicating which subgraph of the diagram must be laid out.
	SplitGraph	Gets or sets a value indicating whether unconnected subgraphs should be laid out independently from each other.
	Stretch	Gets or sets the stretch factor.
	Surface	Specifies on what surface the nodes should be placed initially.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)
	BeginArrange	Initializes the layout object for a step-by-step execution.
	EndArrange	Completes the step-by-step execution started with a call to **BeginArrange**.
	Iterate	Overloaded.

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout Constructor

Initializes a new instance of the SpringLayout class.

Overload List

Name	Description
SpringLayout ()	Initializes a new instance of the SpringLayout class with the default settings.
SpringLayout (Int32, Double, Boolean)	Initializes a new instance of the SpringLayout class with the specified settings.
SpringLayout (Int32, Double, Boolean, Boolean)	Initializes a new instance of the SpringLayout class with the specified settings.

See Also

SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout Constructor ()

Initializes a new instance of the SpringLayout class with the default settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SpringLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SpringLayout Constructor Overload List
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout Constructor (Int32, Double, Boolean)

Initializes a new instance of the SpringLayout class with the specified settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SpringLayout (
    int iterationCount,
    double nodeDistance,
    bool minimizeCrossings
)
```

Visual Basic Copy Code

```
Public New ( _
    iterationCount As Integer, _
    nodeDistance As Double, _
    minimizeCrossings As Boolean _
)
```

Parameters

iterationCount
The number of iterations to perform.

nodeDistance
The distance between nodes.

minimizeCrossings
Specifies whether to try to minimize crossings while layouting.

See Also

SpringLayout Constructor Overload List
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout Constructor (Int32, Double, Boolean, Boolean)

Initializes a new instance of the SpringLayout class with the specified settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SpringLayout (
    int iterationCount,
    double nodeDistance,
    bool minimizeCrossings,
    bool keepGroupLayout
)
```

Visual Basic Copy Code

```
Public New ( _
    iterationCount As Integer, _
    nodeDistance As Double, _
    minimizeCrossings As Boolean, _
    keepGroupLayout As Boolean _
)
```

Parameters

iterationCount
The number of iterations to perform.

nodeDistance
The distance between nodes.

minimizeCrossings
Specifies whether to try to minimize crossings while layouting.

keepGroupLayout
Specifies whether to keep the relative position of grouped items.

See Also

SpringLayout Constructor Overload List
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout Methods

SpringLayout.BeginArrange Method

Initializes the layout object for a step-by-step execution.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool BeginArrange (
    Diagram diagram,
    DiagramItemCollection items
)
```

Visual Basic Copy Code

```
Public Function BeginArrange( _
    diagram As Diagram, _
    items As DiagramItemCollection _
) As Boolean
```

Parameters

diagram
The Diagram that should be arranged.

items
A collection of nodes and links from *diagram* that define the subgraph to arrange.

Return Value

true if the layout object has been initialized successfully; otherwise, **false**.

Remarks

You can use this method together with Iterate and EndArrange to display an animation of the layout process. To run all iterations of the process at once and display just the end result, call Arrange.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.EndArrange Method

Completes the step-by-step execution started with a call to BeginArrange.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void EndArrange ()
```

Visual Basic Copy Code

```
Public Sub EndArrange ()
```

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.Iterate Method

Overload List

Name	Description
SpringLayout.Iterate (Int32, Int32)	Runs one or more layout iterations.
SpringLayout.Iterate (Int32, Int32, Boolean)	Runs one or more layout iterations.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.Iterate Method (Int32, Int32)

Runs one or more layout iterations.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void Iterate (
    int iterationFrom,
    int iterationTo
)
```

Visual Basic Copy Code

```
Public Sub Iterate( _
    iterationFrom As Integer, _
    iterationTo As Integer _
)
```

Parameters

iterationFrom
Specifies from which iteration to start.

iterationTo
Specifies at which iteration to stop.

Remarks

Use this method to create an animation of the layout process, for example, by calling it from a timer. Call BeginArrange to prepare the layout object for use of **Iterate**, and call EndArrange when done. Each call to **Iterate** should start from the iteration next to the one where the previous call has stopped.

See Also

Iterate Method Overload List
SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.Iterate Method (Int32, Int32, Boolean)

Runs one or more layout iterations.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void Iterate (
    int iterationFrom,
    int iterationTo,
    bool asynchronous
)
```

Visual Basic Copy Code

```
Public Sub Iterate( _
    iterationFrom As Integer, _
    iterationTo As Integer, _
    asynchronous As Boolean _
)
```

Parameters

iterationFrom
Specifies from which iteration to start.

iterationTo
Specifies at which iteration to stop.

asynchronous
true to disable UI updates, or false otherwise.

Remarks

Use this method to create an animation of the layout process, for example, by calling it from a timer. Call BeginArrange to prepare the layout object for use of Iterate, and call EndArrange when done. Each call to Iterate starts from the iteration next to the one where the previous call has stopped.

See Also

Iterate Method Overload List
SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout Properties

SpringLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.EnableClusters Property

Gets or sets a value indicating whether to enable allocation of nodes to distinct clusters.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool EnableClusters { get; set; }
```

Visual Basic Copy Code

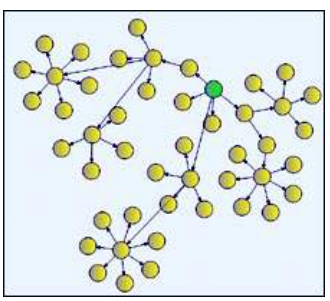
```
Public Property EnableClusters As Boolean
```

Property Value

true to detect and process node clusters; otherwise, **false**. The default is **false**.

Remarks

If enabled, the SpringLayout algorithm detects node clusters and places them at distinct positions. Each cluster consists of a center node surrounded by several local nodes. Clusters might be connected with other clusters by their center nodes. Best-looking results are achieved when there aren't links between the local nodes in a cluster.



See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.IterationCount Property

Gets or sets for how many iterations to run the layout routine.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int IterationCount { get; set; }
```

Visual Basic Copy Code

```
Public Property IterationCount As Integer
```

Property Value

An integer value specifying the number of iterations. The default is 200.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.Layout3D Property

Specifies whether the layout should set the 3D coordinates of nodes for display inside **DiagramView3D**.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool Layout3D { get; set; }
```

Visual Basic Copy Code

```
Public Property Layout3D As Boolean
```

Property Value

Remarks

true to set the 3D coordinates of nodes; otherwise, **false**. The default value is **false**.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace
Surface Property

SpringLayout.LayoutArea Property

Gets or sets the size of the layout area.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Rect LayoutArea { get; set; }
```

Visual Basic Copy Code

```
Public Property LayoutArea As Rect
```

Property Value

A **Microsoft.Maui.Graphics.Rect** representing the boundaries of the layout area. The default is **Empty**.

Remarks

If the layout area is not explicitly set, the algorithm will choose one suitable for the graph size.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.LayoutMethod Property

Specifies what energy model to use for the simulation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SpringLayoutMethod LayoutMethod { get; set; }
```

Visual Basic Copy Code

```
Public Property LayoutMethod As SpringLayoutMethod
```

Property Value

A member of the SpringLayoutMethod enumeration. The default value is **FruchtermanReingold**.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.LinkType Property

Gets or sets the appearance of the links in the final layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SpringLayoutLinkType LinkType { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkType As SpringLayoutLinkType
```

Property Value

A member of the SpringLayoutLinkType enumeration. The default value is **Straight**.

Remarks

If this property is set to Rounded, the layout method sets the links' Shape to Bezier and simulates repulsion forces exerted by their intermediate control points. The **Rounded** option produces better results when LayoutMethod is set to Classic.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.MinimizeCrossings Property

Gets or sets a value indicating whether crossing minimization mode is enabled.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool MinimizeCrossings { get; set; }
```

Visual Basic Copy Code

```
Public Property MinimizeCrossings As Boolean
```

Property Value

true to attempt crossing minimization; otherwise, **false**. The default is **true**.

Remarks

If enabled, SpringLayout tries to decrease number of crossing links. However this could lead to poor symmetry and node distribution. For better results, use greater IterationCount.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Spring layout".

See Also

SpringLayout.NodeDistance Property

Gets or sets the desired distance between nodes.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double NodeDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeDistance As Double
```

Property Value

A **double** value specifying the distance between nodes. The default is 30.

Remarks

The final distance between connected nodes is determined by multiplying the Weight of the link that connects them by the value of **NodeDistance**.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.Randomize Property

Gets or sets a value indicating whether nodes should be placed at random positions when the layout routine starts.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool Randomize { get; set; }
```

Visual Basic Copy Code

```
Public Property Randomize As Boolean
```

Property Value

true to start with random node positions, otherwise **false**. The default value is **false**.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace
RandomSeed Property

SpringLayout.RandomSeed Property

Gets or sets the randomization seed to use for generating random node positions.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int RandomSeed { get; set; }
```

Visual Basic Copy Code

```
Public Property RandomSeed As Integer
```

Property Value

An integer value passed as a seed to the random number generator. The default value is 0.

Remarks

If set to 0, SpringLayout uses the current time as a seed value. Set a value greater than 0 to use a fixed seed. Random numbers are generated when nodes are at the same initial position or when Randomize is enabled. Using a fixed seed helps producing consistent layouts at each run of the Arrange method.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.RepulsionFactor Property

Gets or sets the multiplier applied to repulsion forces between nodes.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double RepulsionFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property RepulsionFactor As Double
```

Property Value

A double value specifying the repulsion factor. The default value is 150.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.Root Property

Gets or sets a node indicating which subgraph of the diagram must be laid out.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public DiagramNode Root { get; set; }
```

Visual Basic Copy Code

```
Public Property Root As DiagramNode
```

Property Value

An instance of the DiagramNode class. The default is a **null** reference (**Nothing** in Visual Basic).

Remarks

All links and nodes that can be reached from the root by following recursively its links from a graph. If the root is specified, only the graph which it defines is laid out by Arrange.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.SplitGraph Property

Gets or sets a value indicating whether unconnected subgraphs should be laid out independently from each other.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool SplitGraph { get; set; }
```

Visual Basic Copy Code

```
Public Property SplitGraph As Boolean
```

Property Value

true if unconnected graphs in the diagram should be laid out independently; otherwise, **false**. The default is **false**.

Remarks

If this property is set to true, each independent graph in the diagram is laid out in its own layout area. Otherwise all subgraphs are arranged in the same layout area. If a Root node is specified and **SplitGraph** is **true**, only the graph that contains the root node will be arranged.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.Stretch Property

Gets or sets the stretch factor.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double Stretch { get; set; }
```

Visual Basic Copy Code

```
Public Property Stretch As Double
```

Property Value

A double value specifying the stretch factor used by the spring layout.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout.Surface Property

Specifies on what surface the nodes should be placed initially.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Surface Surface { get; set; }
```

Visual Basic Copy Code

```
Public Property Surface As Surface
```

Property Value

A member of the **Surface** enumeration. The default value is Sphere.

Remarks

Specifies on what surface the graph nodes should be placed initially, when Layout3D is enabled.

See Also

SpringLayout Members
SpringLayout Class
MindFusion.Diagramming.Layout Namespace

SpringLayout Events

SpringLayoutLinkType Enumeration

Specifies the style of the diagram links after they are laid out.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum SpringLayoutLinkType
```

Visual Basic Copy Code

```
Public Enum SpringLayoutLinkType
```

Members

	Member name	Description
	Straight	Indicates straight links with a single segment.
	Rounded	Indicates Bézier links with a single segment.

See Also

MindFusion.Diagramming.Layout Namespace

SwimlaneLayout Class

Implements an algorithm which arranges nodes in lanes.

Syntax

C# Copy Code

```
public class SwimlaneLayout : Layout
```

Visual Basic Copy Code

```
Public Class SwimlaneLayout
    Inherits Layout
```

Remarks

This layout algorithm can be used to arrange process diagrams in which nodes representing activities are placed in swimlanes representing resources. The index of the resource allocated to an activity should be assigned to the corresponding node's LayoutTraits[SwimlaneLayoutTraits.Lane].

By default, the algorithm works with the diagram's LaneGrid, but its SwimlaneGrid property can be set to any class that implements ISwimlaneGrid. This allows applying the layout to a custom-drawn grid rendered through the DrawBackground event, or one composed of locked background nodes.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.SwimlaneLayout

See Also

SwimlaneLayout Members
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout Members

Implements an algorithm which arranges nodes in lanes.

The following tables list the members exposed by the SwimlaneLayout type.

Public Constructors

	Name	Description
	SwimlaneLayout	Initializes a new instance of the **SwimlaneLayout** class with the default settings.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	CollapseEmptyLanes	Get or sets a value indicating whether lanes without associated nodes will shrink to zero size. By default, empty lanes will have size equal to LaneDistance.
	CompactNodes	Gets or sets a value indicating whether the nodes within individual lanes are compacted.
	CustomSortOrder	Gets or sets a value indicating whether the nodes in a lane should be arranged according to the custom sort order specified through LayoutTraits.
	Direction	Gets or sets the direction of the arranged diagram.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	KeepLaneSizes	Gets or sets a value indicating whether the layout will attempt to preserve the original sizes of the lanes in the supplied grid.
	LaneDistance	Gets or sets the spacing between adjacent lanes in the final layout.
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	NodeDistance	Gets or sets the spacing between adjacent nodes within a lane in the final layout.
	Orientation	Gets or sets the overall orientation of the arranged diagram.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	SwimlaneGrid	Gets or sets the object through which the layout can manipulate the swimlane grid.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.

	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout Constructor

Initializes a new instance of the SwimlaneLayout class with the default settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SwimlaneLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout Properties

SwimlaneLayout.CollapseEmptyLanes Property

Get or sets a value indicating whether lanes without associated nodes will shrink to zero size. By default, empty lanes will have size equal to LaneDistance.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool CollapseEmptyLanes { get; set; }
```

Visual Basic Copy Code

```
Public Property CollapseEmptyLanes As Boolean
```

Property Value

true to collapse empty lanes, otherwise **false**. The default value is **false**.

Remarks

To be able to shrink the lanes of a LaneGrid to zero size, the MinHeaderSize property of the lane grid must be set to an appropriate value.

See Also

SwimlaneLayout Members
SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout.CompactNodes Property

Gets or sets a value indicating whether the nodes within individual lanes are compacted.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool CompactNodes { get; set; }
```

Visual Basic Copy Code

```
Public Property CompactNodes As Boolean
```

Property Value

true to compact nodes, otherwise **false**. The default value is **true**.

See Also

SwimlaneLayout Members
SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout.CustomSortOrder Property

Gets or sets a value indicating whether the nodes in a lane should be arranged according to the custom sort order specified through LayoutTraits.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool CustomSortOrder { get; set; }
```

Visual Basic Copy Code

```
Public Property CustomSortOrder As Boolean
```

Property Value

See Also

SwimlaneLayout Members
SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(20, 20).

See Also

SwimlaneLayout Members
SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout.Direction Property

Gets or sets the direction of the arranged diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Direction Direction { get; set; }
```

Visual Basic Copy Code

```
Public Property Direction As Direction
```

Property Value

A member of the Direction enumeration. The default value is **Straight**.

Remarks

If Orientation is set to Horizontal, Direction.Straight stands for "left to right" and Direction.Reversed for "right to left". If **Orientation** is set to Vertical, Direction.Straight stands for "top to bottom" and Direction.Reversed for "bottom to top".

See Also

SwimlaneLayout Members
SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout.KeepLaneSizes Property

Gets or sets a value indicating whether the layout will attempt to preserve the original sizes of the lanes in the supplied grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool KeepLaneSizes { get; set; }
```

Visual Basic Copy Code

```
Public Property KeepLaneSizes As Boolean
```

Property Value

true to preserve lane sizes, otherwise **false**. The default value is **false**.

See Also

SwimlaneLayout Members
SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout.LaneDistance Property

Gets or sets the spacing between adjacent lanes in the final layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double LaneDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property LaneDistance As Double
```

Property Value

A **double** value specifying the distance between swimlanes. The default value is 20.

Remarks

For each pair of swimlanes, this value specifies the distance between the bottommost node border from the upper swimlane and the topmost node border from the lower swimlane.

See Also

SwimlaneLayout Members
SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Swimlane layout".

See Also

SwimlaneLayout Members
SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout.NodeDistance Property

Gets or sets the spacing between adjacent nodes within a lane in the final layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double NodeDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeDistance As Double
```

Property Value

A **double** value specifying the distance between adjacent nodes in the same swimlane. The default is 25.

See Also

SwimlaneLayout Members
SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout.Orientation Property

Gets or sets the overall orientation of the arranged diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic

```
Public Property Orientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default value is **Horizontal**.

See Also

SwimlaneLayout Members
SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout.SwimlaneGrid Property

Gets or sets the object through which the layout can manipulate the swimlane grid.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public ISwimlaneGrid SwimlaneGrid { get; set; }
```

Visual Basic

```
Public Property SwimlaneGrid As ISwimlaneGrid
```

Property Value

An instance of a class that implements the ISwimlaneGrid interface. The default is **null** (**Nothing** in VB.NET).

Remarks

If this property is set to **null**, the layout will operate on the diagram's LaneGrid.

See Also

SwimlaneLayout Members
SwimlaneLayout Class
MindFusion.Diagramming.Layout Namespace

SwimlaneLayout Events

TopologicalLayout Class

Applies topological ordering to the graph. This layout makes it easy to discern graph features such as overall flow direction, cycles and nested cycles.

Syntax

C#

```
public class TopologicalLayout : Layout
```

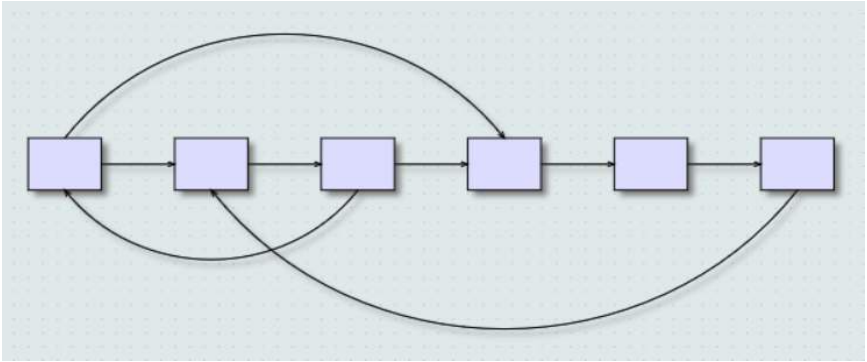
Visual Basic

```
Public Class TopologicalLayout
    Inherits Layout
```

Remarks

Topological ordering of graph nodes is one where link's origin node is placed before link's destination. TopologicalLayout arranges nodes in a row or a column (depending on the value of the Orientation property) in such a way that there are no backward links when the graph is acyclic. If the graph contains cycles, the algorithms selects ordering with as few backward links as possible.

Links that connect non-adjacent nodes are rendered as arcs. Links connecting adjacent nodes can be rendered either as arcs or as straight lines, depending on the BendAdjacentLinks property. The amplitude of link arcs is proportional to the distance between respective connected nodes. All forward links are rendered at one side of the nodes and all back links are rendered at the opposite side. Thus, all bent links follow the same rotation direction.



Example

The following example applies **TopologicalLayout** to the diagram:

C# Copy Code

```
var layout = new TopologicalLayout();
layout.Orientation = MindFusion.Diagramming.Layout.Orientation.Horizontal;
diagram.Arrange(layout);
diagram.ResizeToFitItems(5);
```

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.TopologicalLayout

See Also

TopologicalLayout Members
MindFusion.Diagramming.Layout Namespace

TopologicalLayout Members

Applies topological ordering to the graph. This layout makes it easy to discern graph features such as overall flow direction, cycles and nested cycles.

The following tables list the members exposed by the TopologicalLayout type.

Public Constructors

	Name	Description
	TopologicalLayout	Initializes a new instance of the **TopologicalLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	BendAdjacentLinks	Gets or sets a value indicating whether to bend links connecting adjacent nodes or draw them as straight lines.

	Direction	Gets or sets the direction of the arranged diagram.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.
	Name	Gets a user-friendly name of the layout.
	NodeDistance	Gets or sets the distance between adjacent nodes in the diagram.
	Orientation	Gets or sets the overall orientation of the arranged diagram.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	SplitGraph	Gets or sets a value indicating whether unconnected subgraphs should be laid out independently from each other.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
⚡	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

TopologicalLayout Class
MindFusion.Diagramming.Layout Namespace

TopologicalLayout Constructor

Initializes a new instance of the TopologicalLayout class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TopologicalLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

TopologicalLayout Class
MindFusion.Diagramming.Layout Namespace

TopologicalLayout Properties

TopologicalLayout.BendAdjacentLinks Property

Gets or sets a value indicating whether to bend links connecting adjacent nodes or draw them as straight lines.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool BendAdjacentLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property BendAdjacentLinks As Boolean
```

Property Value

true to bend links connecting adjacent nodes; **false**, to draw them as straight lines. The default is **false**.

Remarks

This property applies only to forward link. Back links connecting adjacent nodes are always rendered as bent.

See Also

TopologicalLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Diagramming.Layout Namespace

TopologicalLayout.Direction Property

Gets or sets the direction of the arranged diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Direction Direction { get; set; }
```

Visual Basic Copy Code

```
Public Property Direction As Direction
```

Property Value

A member of the Direction enumeration. The default is **Straight**.

Remarks

If Orientation is set to Horizontal, Direction.Straight stands for "left to right" and **Direction.Reversed** for "right to left". If **Orientation** is set to **Vertical**, **Direction.Straight** stands for "top to bottom" and **Direction.Reversed** for "bottom to top".

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Diagramming.Layout Namespace

TopologicalLayout.MultipleGraphsPlacement Property

Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public new MultipleGraphsPlacement MultipleGraphsPlacement { get; set; }
```

Visual Basic Copy Code

```
Public Shadows Property MultipleGraphsPlacement As MultipleGraphsPlacement
```

Property Value

A member of the MultipleGraphsPlacement enumeration. The default value is **Vertical**.

Remarks

If the diagram contains several unconnected subgraphs and SplitGraph is set to **true**, each of them is processed by the algorithm independently from the others. This property specifies how such subgraphs should be placed one relatively to another. The possible options are to arrange them in a row, in a column, or in such a way that the diagram covers a minimal area.

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Diagramming.Layout Namespace

TopologicalLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Topological layout".

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Diagramming.Layout Namespace

TopologicalLayout.NodeDistance Property

Gets or sets the distance between adjacent nodes in the diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double NodeDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeDistance As Double
```

Property Value

A **double** value specifying the distance between adjacent nodes. The default is 20.

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Diagramming.Layout Namespace

TopologicalLayout.Orientation Property

Gets or sets the overall orientation of the arranged diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic Copy Code

```
Public Property Orientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default is **Horizontal**.

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Diagramming.Layout Namespace

TopologicalLayout.SplitGraph Property

Gets or sets a value indicating whether unconnected subgraphs should be laid out independently from each other.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool SplitGraph { get; set; }
```

Visual Basic Copy Code

```
Public Property SplitGraph As Boolean
```

Property Value

true if unconnected graphs in the diagram should be laid out independently; otherwise, **false**. The default is **true**.

Remarks

If this property is set to **true**, each independent graph in the diagram is laid out in its own layout area. Otherwise all subgraphs are arranged in the same layout area.

See Also

TopologicalLayout Members
TopologicalLayout Class
MindFusion.Diagramming.Layout Namespace

TopologicalLayout Events

TreeLayout Class

Implements several algorithms for arranging tree structures.

Syntax

C# Copy Code

```
public class TreeLayout : Layout
```

Visual Basic Copy Code

```
Public Class TreeLayout
    Inherits Layout
```

Remarks

Set the Type property of a **TreeLayout** instance to select which tree layout algorithm to run, and call the Arrange method to apply the layout on a tree/subtree in the current diagram.

As long as the graph is a tree, the algorithm can find the root node automatically. You can override it by setting the Root property, e.g. in order to arrange only a selected branch of the tree. The distance between the root and the second level and between all subsequent levels can be set through the LevelDistance property. The distance between adjacent nodes in the same level is set through NodeDistance.

In addition to arranging the tree nodes, **TreeLayout** also sets the shape of links as set through its LinkStyle property. Anchoring specifies whether the links should be attached to the anchor points of tree nodes. Note that links that are not part of the tree are ignored, e.g. graph links that go in opposite direction of the general parent-to-child flow or cross-links that connect different tree branches. The tree layout algorithm produces best results if it can place link ends at arbitrary positions over nodes' outlines, that is, when using the Ignore value. If your application logic requires links to be always attached to anchor points, use the other two values. In such cases **Reassign** would yield better results. You can post-process such links after calling **Arrange**, for example by setting their ControlPoints coordinates or by calling their Route method.

By default, the order of child nodes arranged under a parent corresponds to the order of links in the parent's **OutgoingLinks** collection. The Balance property can be set to Preserve to keep the original geometric order of child nodes when arranging them under their parent. This can be used to arrange tree branches in a specific order by first placing them in increasing horizontal or vertical positions. Another possibility is to apply **TreeLayout** from NodeModified handler to let users rearrange the tree branches by dragging and dropping a node.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.TreeLayout

See Also

TreeLayout Members
MindFusion.Diagramming.Layout Namespace
TreeMapLayout Class

TreeLayout Members

Implements several algorithms for arranging tree structures.

The following tables list the members exposed by the TreeLayout type.

Public Constructors

	Name	Description
	TreeLayout	Overloaded. Initializes a new instance of the **TreeLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	Balance	Gets or sets a value indicating the ordering of the child nodes in order to produce layouts with custom balance.
	CompactAssistants	Gets or sets a value indicating whether the "assistant" nodes on the same side of a single parent are arranged as close enough to each other as possible. Turning this off will force the nodes on one side to be aligned with their counterparts in the other and occasionally leaving space between adjacent branches.
	Direction	Gets or sets the tree direction.
	EnableAssistants	Gets or sets a value indicating whether the "assistant" trait is regarded when performing the layout.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	IgnoreLinkDirection	Gets or sets a value indicating whether the tree layout algorithm should ignore the direction of links when arranging a tree.
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	KeepRootPosition	Gets or sets a value indicating whether to keep the root node at its original position.
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	LevelDistance	Gets or sets how much space to leave between levels of the tree.
	LinkOffset	Gets or sets the links' offset from the centers of the nodes. This property is applicable only for Centered tree layouts.
	LinkStyle	Gets or sets a value indicating how links in the arranged tree should look.
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.

	NodeDistance	Gets or sets how much space to leave between nodes on the same level.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	ReversedLinks	Gets or sets a value indicating the link direction that relates parent nodes to child nodes.
	Root	Gets or sets the root node of the tree.
	StretchFactor	Gets or sets the stretch factor applied on Radial tree layouts.
	Type	Gets or sets what type of tree layout to apply to the diagram.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)
	UpdateLink	Updates the style and position of the specified link.

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout Constructor

Initializes a new instance of the TreeLayout class.

Overload List

Name	Description

TreeLayout ()	Initializes a new instance of the **TreeLayout** class with the default settings.
TreeLayout (DiagramNode)	Initializes a new instance of the **TreeLayout** class with the specified node as root.
TreeLayout (DiagramNode, TreeLayoutType)	Initializes a new instance of the **TreeLayout** class with the specified node as root and the specified type.
TreeLayout (DiagramNode, TreeLayoutType, Boolean, TreeLayoutLinkType, TreeLayoutDirections, Double, Double, Boolean, Size)	Initializes a new instance of the **TreeLayout** class with the specified settings.
TreeLayout (DiagramNode, TreeLayoutType, Boolean, TreeLayoutLinkType, TreeLayoutDirections, Double, Double, Boolean, Size, Boolean)	Initializes a new instance of the **TreeLayout** class with the specified settings.
TreeLayout (TreeLayoutType)	Initializes a new instance of the **TreeLayout** class with the specified type.

See Also

TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout Constructor ()

Initializes a new instance of the TreeLayout class with the default settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TreeLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

TreeLayout Constructor Overload List
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout Constructor (DiagramNode)

Initializes a new instance of the TreeLayout class with the specified node as root.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TreeLayout (
    DiagramNode root
)
```

Visual Basic Copy Code

```
Public New ( _
    root As DiagramNode _
)
```

Parameters

root
The root of the arranged tree.

See Also

TreeLayout Constructor Overload List
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout Constructor (DiagramNode, TreeLayoutType)

Initializes a new instance of the TreeLayout class with the specified node as root and the specified type.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TreeLayout (
    DiagramNode root,
    TreeLayoutType type
)
```

Visual Basic Copy Code

```
Public New ( _
    root As DiagramNode, _
    type As TreeLayoutType _
)
```

Parameters

root
The root of the arranged tree.

type
The type of the tree layout.

See Also

TreeLayout Constructor Overload List
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout Constructor (DiagramNode, TreeLayoutType, Boolean, TreeLayoutLinkType, TreeLayoutDirections, Double, Double, Boolean, Size)

Initializes a new instance of the TreeLayout class with the specified settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TreeLayout (
    DiagramNode root,
    TreeLayoutType type,
    bool reversedLinks,
    TreeLayoutLinkType linkStyle,
    TreeLayoutDirections direction,
    double levelDistance,
    double nodeDistance,
    bool keepRootPosition,
    Size margins
)
```

Visual Basic Copy Code

```
Public New ( _
    root As DiagramNode, _
    type As TreeLayoutType, _
    reversedLinks As Boolean, _
    linkStyle As TreeLayoutLinkType, _
    direction As TreeLayoutDirections, _
    levelDistance As Double, _
    nodeDistance As Double, _
    keepRootPosition As Boolean, _
    margins As Size _
)
```

Parameters

root
The root of the arranged tree.

type
The type of the tree layout.

reversedLinks
Specifies whether the parent-to-child direction in the tree is reversed, that is, from the end of the link towards its beginning.

linkStyle
The style of the links in the produced layout.

direction
The layout direction.

levelDistance
The distance between adjacent levels.

nodeDistance
The distance between adjacent nodes.

keepRootPosition
Specifies whether to keep the position of the tree root.

margins
The margin between the produced tree and the left side of the diagram bounding rectangle.

See Also

TreeLayout Constructor Overload List
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout Constructor (DiagramNode, TreeLayoutType, Boolean, TreeLayoutLinkType, TreeLayoutDirections, Double, Double, Boolean, Size, Boolean)

Initializes a new instance of the TreeLayout class with the specified settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TreeLayout (
    DiagramNode root,
    TreeLayoutType type,
    bool reversedLinks,
    TreeLayoutLinkType linkStyle,
    TreeLayoutDirections direction,
    double levelDistance,
    double nodeDistance,
    bool keepRootPosition,
    Size margins,
    bool keepGroupLayout
)
```

Visual Basic Copy Code

```
Public New ( _
    root As DiagramNode, _
    type As TreeLayoutType, _
    reversedLinks As Boolean, _
    linkStyle As TreeLayoutLinkType, _
    direction As TreeLayoutDirections, _
    levelDistance As Double, _
    nodeDistance As Double, _
    keepRootPosition As Boolean, _
    margins As Size, _
    keepGroupLayout As Boolean _
)
```

Parameters

root
The root of the arranged tree.

type
The type of the tree layout.

reversedLinks
Specifies whether the parent-to-child direction in the tree should be reversed, that is, from the end of the link towards its beginning.

linkStyle
The style of the links in the produced layout.

direction
The layout direction.

levelDistance
The distance between adjacent levels.

nodeDistance
The distance between adjacent nodes.

keepRootPosition
Specifies whether to keep the position of the tree root.

margins
The margin between the produced tree and the left side of the diagram bounding rectangle.

keepGroupLayout
Specifies whether to keep the relative position of grouped items.

See Also

TreeLayout Constructor Overload List
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout Constructor (TreeLayoutType)

Initializes a new instance of the TreeLayout class with the specified type.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TreeLayout (
    TreeLayoutType type
)
```

Visual Basic Copy Code

```
Public New ( _
    type As TreeLayoutType _
)
```

Parameters

type
The type of the tree layout.

See Also

TreeLayout Constructor Overload List
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout Methods

TreeLayout.UpdateLink Method

Updates the style and position of the specified link.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void UpdateLink (
    DiagramLink a,
    bool ignoredDirection
)
```

Visual Basic Copy Code

```
Public Sub UpdateLink( _
    a As DiagramLink, _
    ignoredDirection As Boolean _
)
```

Parameters

a
A DiagramLink that will be updated.

ignoredDirection

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout Properties

TreeLayout.Balance Property

Gets or sets a value indicating the ordering of the child nodes in order to produce layouts with custom balance.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TreeLayoutBalance Balance { get; set; }
```

Visual Basic Copy Code

```
Public Property Balance As TreeLayoutBalance
```

Property Value

A member of the TreeLayoutBalance enumeration. The default value is **Original**.

Remarks

The LeftHeavy, **Original** and **RightHeavy** values are considered when Type is set to HorizontalVertical. The **Preserve** and **Reversed** values are considered by all other tree layout types, and will lead to sorting tree branches by the original horizontal, vertical or angular positions of nodes.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.CompactAssistants Property

Gets or sets a value indicating whether the "assistant" nodes on the same side of a single parent are arranged as close enough to each other as possible. Turning this off will force the nodes on one side to be aligned with their counterparts in the other and occasionally leaving space between adjacent branches.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool CompactAssistants { get; set; }
```

Visual Basic Copy Code

```
Public Property CompactAssistants As Boolean
```

Property Value

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.Direction Property

Gets or sets the tree direction.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TreeLayoutDirections Direction { get; set; }
```

Visual Basic Copy Code

```
Public Property Direction As TreeLayoutDirections
```

Property Value

A member of the TreeLayoutDirections enumeration.

Remarks

This property must be set to a combination of two TreeLayoutDirections members when using Type is HorizontalVertical.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.EnableAssistants Property

Gets or sets a value indicating whether the "assistant" trait is regarded when performing the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool EnableAssistants { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableAssistants As Boolean
```

Property Value

true to enable assistant nodes, otherwise **false**; the default value is **false**.

Remarks

Enabling this property lets you create organizational charts as in MS office. To designate some nodes as assistants, assign **true** as a value for the TreeLayoutTraits.Assistant key in the node's LayoutTraits.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.IgnoreLinkDirection Property

Gets or sets a value indicating whether the tree layout algorithm should ignore the direction of links when arranging a tree.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool IgnoreLinkDirection { get; set; }
```

Visual Basic Copy Code

```
Public Property IgnoreLinkDirection As Boolean
```

Property Value

true to ignore link direction; otherwise, **false**. The default is **false**.

Remarks

Parent-child direction is considered the one in which nodes go further from the root, instead that of the links. If **IgnoreLinkDirection** is set to **true**, you must explicitly specify the tree Root.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.KeepRootPosition Property

Gets or sets a value indicating whether to keep the root node at its original position.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool KeepRootPosition { get; set; }
```

Visual Basic

```
Public Property KeepRootPosition As Boolean
```

Property Value

true to keep the root position; otherwise, **false**.

Remarks

If the root position is not kept, that is, the value of this property is **false**, the entire tree is dispositioned after the layout is complete, according to the values of Margins.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.LevelDistance Property

Gets or sets how much space to leave between levels of the tree.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public double LevelDistance { get; set; }
```

Visual Basic

```
Public Property LevelDistance As Double
```

Property Value

A **double** value specifying the distance between adjacent levels in the tree.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.LinkOffset Property

Gets or sets the links' offset from the centers of the nodes. This property is applicable only for Centered tree layouts.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public double LinkOffset { get; set; }
```

Visual Basic

```
Public Property LinkOffset As Double
```

Property Value

A **double** value specifying offset of links from the parent node's center. The default value is 0.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.LinkStyle Property

Gets or sets a value indicating how links in the arranged tree should look.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TreeLayoutLinkType LinkStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property LinkStyle As TreeLayoutLinkType
```

Property Value

A member of the TreeLayoutLinkType enumeration.

Remarks

This property is ignored if Type is set to Radial.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Tree layout".

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.NodeDistance Property

Gets or sets how much space to leave between nodes on the same level.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double NodeDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeDistance As Double
```

Property Value

A **double** value specifying the distance between adjacent nodes.

Remarks

This property is ignored if Type is set to Radial.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.ReversedLinks Property

Gets or sets a value indicating the link direction that relates parent nodes to child nodes.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool ReversedLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property ReversedLinks As Boolean
```

Property Value

true to indicate reversed direction, that is, the parent node is the one the link points to; otherwise, **false**.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.Root Property

Gets or sets the root node of the tree.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public DiagramNode Root { get; set; }
```

Visual Basic

```
Public Property Root As DiagramNode
```

Property Value

An instance of the DiagramNode class that represents the root node.

Remarks

If not set, MvcDiagram will try to find the root automatically. This root must be specified if IgnoreLinkDirection is set to **true**.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.StretchFactor Property

Gets or sets the stretch factor applied on Radial tree layouts.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public double StretchFactor { get; set; }
```

Visual Basic

```
Public Property StretchFactor As Double
```

Property Value

A real number specifying the ratio between the horizontal and vertical radii of radial layout outlines. The default is 1.

Remarks

When applying a radial layout, node layers are laid out in concentric circles around the Root node. Stretching them by changing this property makes layers laid out in elliptic outlines.

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout.Type Property

Gets or sets what type of tree layout to apply to the diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public TreeLayoutType Type { get; set; }
```

Visual Basic

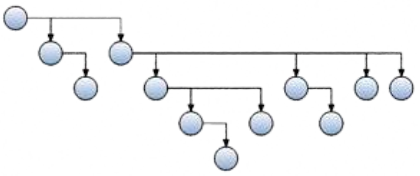
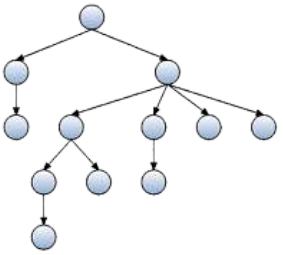
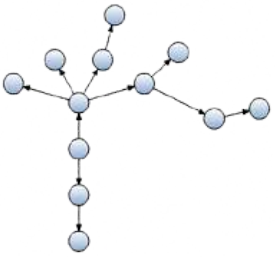
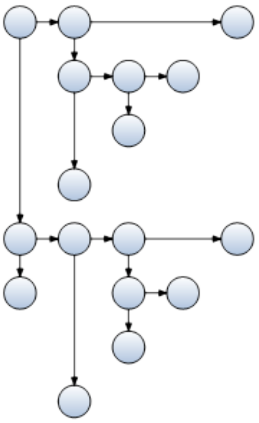
```
Public Property Type As TreeLayoutType
```

Property Value

A member of the TreeLayoutType enumeration.

Remarks

The available tree layout types include the following:

Layout type	Appearance
Cascading	
Centered	
Radial	
HorizontalVertical	

See Also

TreeLayout Members
TreeLayout Class
MindFusion.Diagramming.Layout Namespace

TreeLayout Events

TreeLayoutBalance Enumeration

Specifies how to order child nodes in order to produce tree layouts with custom balance.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum TreeLayoutBalance
```

Visual Basic Copy Code

```
Public Enum TreeLayoutBalance
```

Members

	Member name	Description
	Original	Indicates that the child order should not be changed.
	Reversed	Indicates that the child order should be reversed.
	LeftHeavy	Indicates that the child nodes with most total children will be processed first, thus producing a left-heavy layout.
	RightHeavy	Indicates that the child nodes with least total children will be processed first, thus producing a right-heavy layout.
	Preserve	Indicates that the layout will attempt to preserve the initial relative ordering of nodes.

See Also

MindFusion.Diagramming.Layout Namespace

TreeLayoutDirections Enumeration

Specifies the direction of a tree layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum TreeLayoutDirections
```

Visual Basic Copy Code

```
Public Enum TreeLayoutDirections
```

Members

	Member name	Description
	TopToBottom	Indicates a top-to-bottom direction.
	BottomToTop	Indicates a bottom-to-top direction.
	LeftToRight	Indicates a left-to-right direction.
	RightToLeft	Indicates a right-to-left direction.

See Also

MindFusion.Diagramming.Layout Namespace

TreeLayoutLinkType Enumeration

Specifies the style of the diagram links after they are laid out.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum TreeLayoutLinkType
```

Visual Basic Copy Code

```
Public Enum TreeLayoutLinkType
```

Members

	Member name	Description
	Straight	Indicates straight links with a single segment.
	Cascading2	Indicates cascading links with two segments.
	Cascading3	Indicates cascading links with three segments.
	Rounded	Indicates Bézier links with a single segment.

See Also

MindFusion.Diagramming.Layout Namespace

TreeLayoutType Enumeration

Specifies the type of a tree layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum TreeLayoutType
```

Visual Basic Copy Code

```
Public Enum TreeLayoutType
```

Members

	Member name	Description
	Cascading	Indicates cascading tree layout.
	Centered	Indicates centered tree layout.
	Radial	Indicates radial tree layout.
	HorizontalVertical	Indicates the Horizontal-Vertical layout algorithm, used to arrange full binary trees.

See Also

MindFusion.Diagramming.Layout Namespace

TreeMapLayout Class

Implements a tree map layout algorithm.

Syntax

C# Copy Code

```
public class TreeMapLayout : Layout
```

Visual Basic Copy Code

```
Public Class TreeMapLayout
    Inherits Layout
```

Remarks

Tree maps represent hierarchies by nesting child nodes within their parents, where the areas of leaf nodes are proportional to their Weight values. Unlike other layout algorithms, **TreeMapLayout** expects hierarchies to be defined via grouping or containment (see AttachTo method and ContainerNode class), and will ignore any links in the diagram.

The diagram area covered by the topmost nodes in a hierarchy is specified via the LayoutArea property. By default, the layout tries to keep the ratio of node sides as close as possible to one. However this could make it hard to distinguish separate levels of the hierarchy. To alleviate that, set Squarify to **false**, and child nodes will be arranged either as a row or a column inside their parent node, alternating directions for each level. The drawback is that when **Weight** ratios differ greatly or nodes contain many children, some nodes could end up with very narrow rectangles.

The padding size between parent nodes and their outermost children is specified via the ContainerPadding property. The padding size between child nodes is specified via Padding.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.TreeMapLayout

See Also

TreeMapLayout Members
MindFusion.Diagramming.Layout Namespace

TreeMapLayout Members

Implements a tree map layout algorithm.

The following tables list the members exposed by the TreeMapLayout type.

Public Constructors

	Name	Description
	TreeMapLayout	Initializes a new instance of the TreeMapLayout class with the default settings.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	ContainerPadding	Gets or sets the padding inside containers.

	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutArea	Gets or sets the rectangle in which the layout will try to arrange the nodes, or null, to use the diagram bounds.
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	Orientation	Gets or sets the orientation of the arranged diagram.
	Padding	Gets or sets the distance between adjacent nodes.
	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)
	Squarify	Gets or sets a value indicating whether the layout should attempt to keep the dimension ratio of the nodes closer to 1.

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

TreeMapLayout Class
MindFusion.Diagramming.Layout Namespace

TreeMapLayout Constructor

Initializes a new instance of the TreeMapLayout class with the default settings.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TreeMapLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

TreeMapLayout Class
MindFusion.Diagramming.Layout Namespace

TreeMapLayout Properties

TreeMapLayout.ContainerPadding Property

Gets or sets the padding inside containers.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double ContainerPadding { get; set; }
```

Visual Basic Copy Code

```
Public Property ContainerPadding As Double
```

Property Value

A **double** value specifying the padding size between parent nodes and their outermost children. The default value is 0.

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Diagramming.Layout Namespace
Padding Property

TreeMapLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Diagramming.Layout Namespace

TreeMapLayout.LayoutArea Property

Gets or sets the rectangle in which the layout will try to arrange the nodes, or null, to use the diagram bounds.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Nullable<Rect> LayoutArea { get; set; }
```

Visual Basic Copy Code

```
Public Property LayoutArea As Nullable(Of Rect)
```

Property Value

A **Rect?** specifying the area where topmost nodes in the hierarchy are laid out. The default value is **null**.

Remarks

If this property is not set, the algorithm uses the value of diagram's Bounds property as target layout rectangle.

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Diagramming.Layout Namespace

TreeMapLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public override string Name { get; }
```

Visual Basic

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Tree map layout".

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Diagramming.Layout Namespace

TreeMapLayout.Orientation Property

Gets or sets the orientation of the arranged diagram.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public Orientation Orientation { get; set; }
```

Visual Basic

```
Public Property Orientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default value is **Vertical**.

Remarks

This value is used only when Squarify is set to **false**, and specifies the orientation of the topmost level of the hierarchy. The algorithm alternates the orientation of each subsequent level, arranging its nodes either in rows or columns.

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Diagramming.Layout Namespace

TreeMapLayout.Padding Property

Gets or sets the distance between adjacent nodes.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#

```
public double Padding { get; set; }
```

Visual Basic

```
Public Property Padding As Double
```

Property Value

A **double** value specifying the padding size between sibling nodes. The default value is 1.

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Diagramming.Layout Namespace
ContainerPadding Property

TreeMapLayout.Squarify Property

Gets or sets a value indicating whether the layout should attempt to keep the dimension ratio of the nodes closer to 1.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool Squarify { get; set; }
```

Visual Basic Copy Code

```
Public Property Squarify As Boolean
```

Property Value

true to keep the ratio of node sides as close as possible to one, or **false** otherwise. The default value is **true**.

Remarks

If set to **false**, each level of the hierarchy is arranged in rows or columns with same height/width, which might create very narrow rectangles when Weight values differ greatly or nodes have many children. The alternating orientation between levels however helps to distinguish between different levels in the hierarchy. The initial direction of nodes (in topmost level) depends on the value of Orientation.

See Also

TreeMapLayout Members
TreeMapLayout Class
MindFusion.Diagramming.Layout Namespace

TreeMapLayout Events

TriangularLayout Class

Performs triangular layout on simple, connected graphs.

Syntax

C# Copy Code

```
public class TriangularLayout : Layout
```

Visual Basic Copy Code

```
Public Class TriangularLayout
    Inherits Layout
```

Remarks

TriangularLayout places nodes on a virtual grid, such that if the source graph is planar, all links are guaranteed to have a single segment and not intersect. If the graph is not planar, its intersecting links can be optionally segmented and bended in order to improve readability. The layout method places the nodes from the external face on a triangle and recursively adds the rest of the nodes as vertices of internal triangles. As a result, it is very effective for near maximal-planar (a.k.a. triangular) graphs.

Inheritance Hierarchy

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      MindFusion.Diagramming.Layout.Layout
        MindFusion.Diagramming.Layout.TriangularLayout

See Also

TriangularLayout Members
MindFusion.Diagramming.Layout Namespace

TriangularLayout Members

Performs triangular layout on simple, connected graphs.

The following tables list the members exposed by the TriangularLayout type.

Public Constructors

	Name	Description
	TriangularLayout	Initializes a new instance of the **TriangularLayout** class.

Public Properties

	Name	Description
	Anchoring	Gets or sets a value indicating how to align links to the anchor points of graph nodes. (Inherited from Layout.)
	BendNonplanarLinks	Gets or sets a value indicating whether the intersecting links in a non-planar graphs will be segmented and bended to improve presentation.
	EnableParallelism	Gets or sets a value indicating whether to arrange subgraphs on separate threads. (Inherited from Layout.)
	GridSize	Gets or sets the size of the virtual grid the layout will place vertices on.
	GrowToFit	Gets or sets a value indicating whether the diagram boundaries should grow to fit the arranged content. (Inherited from Layout.)
	KeepGroupLayout	Gets or sets a value indicating whether to treat each Group of nodes as a single vertex in the arranged graph. (Inherited from Layout.)
	LayoutLink	Gets or sets a delegate, which is invoked when the position of a link is changed by the layout. (Inherited from Layout.)
	LayoutNode	Gets or sets a delegate, which is invoked when the position of a node is changed by the layout. (Inherited from Layout.)
	Margins	Gets or sets the distance between the top-left corner of the smallest rectangle containing the arranged diagram and the top-left point of the document bounds. (Inherited from Layout.)
	MaxDegreeOfParallelism	Gets or sets the maximum degree of parallelism. (Inherited from Layout.)
	MultipleGraphsPlacement	Gets or sets a value indicating how multiple independent graphs in the diagram should be positioned relatively to each other. (Inherited from Layout.)
	Name	Gets a user-friendly name of the layout.
	Orientation	Gets or sets the orientation of the arranged graph.

	Progress	Gets or sets a delegate, which is invoked regularly during the layout process to inform clients about the layout progress. (Inherited from Layout.)
	RepeatingLinksPadding	Specifies the padding distance between repeating links. (Inherited from Layout.)

Protected Properties

	Name	Description
	DefaultMarginValue	Gets the default value of the Margins property.
	SuppressCompleteEvent	Specifies whether raising the LayoutComplete event should be suppressed for this layout instance. (Inherited from Layout.)

Public Methods

	Name	Description
	Arrange	Overloaded. Applies the layout to a Diagram instance. (Inherited from Layout.)
	ArrangeAnimated	Overloaded. (Inherited from Layout.)
	ArrangeAsync	Overloaded. (Inherited from Layout.)

Protected Methods

	Name	Description
	OnLayoutComplete	Raises the LayoutComplete event. (Inherited from Layout.)

Public Events

	Name	Description
	LayoutComplete	Raised when a layout is finished arranging a graph. (Inherited from Layout.)

See Also

TriangularLayout Class
MindFusion.Diagramming.Layout Namespace

TriangularLayout Constructor

Initializes a new instance of the TriangularLayout class.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public TriangularLayout ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

TriangularLayout Properties

TriangularLayout.BendNonplanarLinks Property

Gets or sets a value indicating whether the intersecting links in a non-planar graphs will be segmented and bended to improve presentation.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool BendNonplanarLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property BendNonplanarLinks As Boolean
```

Property Value

true to bend links, otherwise **false**. The default value is **false**.

See Also

TriangularLayout Members
TriangularLayout Class
MindFusion.Diagramming.Layout Namespace

TriangularLayout.DefaultMarginValue Property

Gets the default value of the Margins property.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
protected override Size DefaultMarginValue { get; }
```

Visual Basic Copy Code

```
Protected Overrides ReadOnly Property DefaultMarginValue As Size
```

Property Value

Returns new Size(5, 5).

See Also

TriangularLayout Members
TriangularLayout Class
MindFusion.Diagramming.Layout Namespace

TriangularLayout.GridSize Property

Gets or sets the size of the virtual grid the layout will place vertices on.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double GridSize { get; set; }
```

Visual Basic Copy Code

```
Public Property GridSize As Double
```

Property Value

The default value is 20.

See Also

TriangularLayout Members
TriangularLayout Class
MindFusion.Diagramming.Layout Namespace

TriangularLayout.Name Property

Gets a user-friendly name of the layout.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string Name { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property Name As String
```

Property Value

Returns "Triangular layout".

See Also

TriangularLayout Members
TriangularLayout Class
MindFusion.Diagramming.Layout Namespace

TriangularLayout.Orientation Property

Gets or sets the orientation of the arranged graph.

Namespace: MindFusion.Diagramming.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic Copy Code

```
Public Property Orientation As Orientation
```

Property Value

A member of the Orientation enumeration. The default value is **Vertical**.

See Also

TriangularLayout Members
TriangularLayout Class
MindFusion.Diagramming.Layout Namespace

TriangularLayout Events

MindFusion.Diagramming.Mvc Namespace

CompositeNodeConverter Class

Serializes CompositeNode instances.

Syntax

C# Copy Code

```
public class CompositeNodeConverter : DiagramNodeConverter
```

Visual Basic Copy Code

```
Public Class CompositeNodeConverter
    Inherits DiagramNodeConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.DiagramItemConverter
      MindFusion.Diagramming.Mvc.DiagramNodeConverter
        MindFusion.Diagramming.Mvc.CompositeNodeConverter

See Also

CompositeNodeConverter Members
MindFusion.Diagramming.Mvc Namespace

CompositeNodeConverter Members

Serializes CompositeNode instances.

The following tables list the members exposed by the CompositeNodeConverter type.

Public Constructors

	Name	Description
	CompositeNodeConverter	Creates a new instance of the CompositeNodeConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

CompositeNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

CompositeNodeConverter Constructor

Creates a new instance of the CompositeNodeConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public CompositeNodeConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

CompositeNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

CompositeNodeConverter Methods

CompositeNodeConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
```

```
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

CompositeNodeConverter Members
CompositeNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

CompositeNodeConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

CompositeNodeConverter Members
CompositeNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

CompositeNodeConverter Properties

CompositeNodeConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

CompositeNodeConverter Members
CompositeNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ContainerNodeConverter Class

Serializes ContainerNode instances.

Syntax

C# Copy Code

```
public class ContainerNodeConverter : DiagramNodeConverter
```

Visual Basic Copy Code

```
Public Class ContainerNodeConverter
    Inherits DiagramNodeConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.DiagramItemConverter
      MindFusion.Diagramming.Mvc.DiagramNodeConverter
        MindFusion.Diagramming.Mvc.ContainerNodeConverter

See Also

ContainerNodeConverter Members
MindFusion.Diagramming.Mvc Namespace

ContainerNodeConverter Members

Serializes ContainerNode instances.

The following tables list the members exposed by the ContainerNodeConverter type.

Public Constructors

	Name	Description
	ContainerNodeConverter	Creates a new instance of the ContainerNodeConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

ContainerNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ContainerNodeConverter Constructor

Creates a new instance of the ContainerNodeConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ContainerNodeConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

ContainerNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ContainerNodeConverter Methods

ContainerNodeConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
```

```
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

ContainerNodeConverter Members
ContainerNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ContainerNodeConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

ContainerNodeConverter Members
ContainerNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ContainerNodeConverter Properties

ContainerNodeConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

ContainerNodeConverter Members
ContainerNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ControlNode Class

Represents diagram nodes which can display custom HTML content.

Syntax

C#

```
public class ControlNode : DiagramNode
```

Visual Basic

```
Public Class ControlNode
    Inherits DiagramNode
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.DiagramItem
    MindFusion.Diagramming.DiagramNode
      MindFusion.Diagramming.Mvc.ControlNode

See Also

ControlNode Members
MindFusion.Diagramming.Mvc Namespace

ControlNode Members

Represents diagram nodes which can display custom HTML content.

The following tables list the members exposed by the ControlNode type.

Public Constructors

	Name	Description
	ControlNode	Overloaded.

Protected Fields

	Name	Description
	disposed	For internal use. (Inherited from DiagramItem.)

Public Properties

	Name	Description
	AllowIncomingLinks	Gets or sets a value indicating whether users are allowed to draw incoming links to this node. (Inherited from DiagramNode.)
	AllowOutgoingLinks	Gets or sets a value indicating whether users are allowed to draw outgoing links from this node. (Inherited from DiagramNode.)
	AnchorPattern	Gets or sets the anchor points to which links are attached when connected to the node. (Inherited from DiagramNode.)
	Bounds	Gets or sets the rectangle that defines the position of the diagram node. (Inherited from DiagramNode.)
	Brush	Gets or sets the brush used to fill the interior of this item. (Inherited from DiagramItem.)
	Constraints	Allows applying constraints to the location and size of the node. (Inherited from DiagramNode.)
	Container	Gets the parent ContainerNode containing this node. (Inherited from DiagramNode.)
	CssClass	Gets or sets the CSS class associated with this item. (Inherited from DiagramItem.)
	EffectiveBrush	Gets the effective brush of the item. (Inherited from DiagramItem.)
	EffectiveFont	Gets the currently effective font of the item. (Inherited from DiagramItem.)
	EffectivePen	Gets the effective pen of the item. (Inherited from DiagramItem.)
	EffectiveShadowBrush	Gets the effective shadow brush of the item. (Inherited from DiagramItem.)
	EffectiveTextBrush	Gets the effective text brush of the item. (Inherited from DiagramItem.)
	Effects	Gets a collection with all effects applied to this node. (Inherited from DiagramNode.)
	EnabledHandles	Gets or sets the kinds of modifications that end-users are permitted to perform on the node. (Inherited from DiagramNode.)
	Expandable	Gets or sets a value indicating whether users are allowed to expand or collapse the subtree that starts from this node. (Inherited from DiagramNode.)
	Expanded	Gets or sets a value indicating whether the subtree starting from this node is expanded. (Inherited from DiagramNode.)
	Font	Gets or sets the font used to render the item's text. (Inherited from DiagramItem.)
	HandlesStyle	Gets or sets a value indicating how the node adjustment handles behave and what do they look like. (Inherited from DiagramNode.)
	HyperLink	Gets or sets the hyperlink associated with this diagram item. (Inherited from DiagramItem.)
	Id	Gets or sets user-defined identifier associated with this item. (Inherited from DiagramItem.)
	IgnoreLayout	Gets or sets a value indicating whether the position of this item should not be changed by automatic layout methods. (Inherited from DiagramItem.)
	IncomingLinks	Gets a collection of links that point to the node. (Inherited from DiagramNode.)
	Labels	Returns the list of supplementary labels displayed by this node. (Inherited from DiagramNode.)
	LayerIndex	Gets or sets the zero-based index of the layer this item is associated with, or -1, if the item is not associated with a layer. (Inherited from DiagramItem.)

	LayoutTraits	Contains layout properties of the node, used by some layout algorithms and by **DiagramView3D**. (Inherited from DiagramItem.)
	Locked	Gets or sets a value indicating whether users are allowed to edit the diagram item. (Inherited from DiagramItem.)
	MasterGroup	Gets a reference to the group to which this node is attached. (Inherited from DiagramNode.)
	MasterItem	Returns the node's **Container** or main item of **MasterGroup**. (Inherited from DiagramNode.)
	Obstacle	Gets or sets a value indicating whether this node is considered an obstacle by the link-routing algorithm. (Inherited from DiagramNode.)
	Opacity	Gets or sets the opacity of this item. (Inherited from DiagramItem.)
	OutgoingLinks	Gets a collection of links that start from this node. (Inherited from DiagramNode.)
	Parent	Gets the diagram that contains this item. (Inherited from DiagramItem.)
	Pen	Gets or sets the pen used to paint the frame of the item. (Inherited from DiagramItem.)
	Printable	Gets or sets a value indicating whether to print the item. (Inherited from DiagramItem.)
	RotationAngle	Gets or sets the rotation angle of this node. (Inherited from DiagramNode.)
	RuntimeIndex	For internal use. (Inherited from DiagramItem.)
	Selected	Gets or sets a value indicating whether a diagram item is selected. (Inherited from DiagramItem.)
	ShadowBrush	Gets or sets the brush used to paint the item's shadow. (Inherited from DiagramItem.)
	ShadowOffsetX	Gets or sets the horizontal offset of the shadow of this item. (Inherited from DiagramItem.)
	ShadowOffsetY	Gets or sets the vertical offset of the shadow of this item. (Inherited from DiagramItem.)
	Style	Gets or sets the style associated with the item. (Inherited from DiagramItem.)
	SubordinateGroup	Gets a reference to the group of items that are attached to this one. (Inherited from DiagramItem.)
	Tag	Gets or sets user-defined data associated with this item. (Inherited from DiagramItem.)
	Template	Gets or sets the HTML string representing the DOM content rendered in this node.
	Text	Gets or sets the text displayed by this item. (Inherited from DiagramItem.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItem.)
	TextFormat	Gets or sets the alignment of text displayed inside this item. (Inherited from DiagramItem.)
	TextPadding	Gets or sets the spacing between the item boundaries and its text. (Inherited from DiagramItem.)
	ToolTip	Gets or sets a short text message displayed when the mouse pointer rests over the diagram element. (Inherited from DiagramItem.)
	TopLevel	DiagramItem.TopLevel override. (Inherited from DiagramNode.)

	Visible	Gets or sets a value indicating whether this item is visible. (Inherited from DiagramItem.)
	Weight	Gets or sets a weight value used in layout and path-finding algorithms. (Inherited from DiagramItem.)
	ZIndex	Gets or sets the z-order position of this item. (Inherited from DiagramItem.)

Protected Properties

	Name	Description
	HasItems	Determines whether the node supports connections to its constituent elements. (Inherited from DiagramNode.)
	MeasureUnit	Returns the diagram measure unit. (Inherited from DiagramItem.)

Public Methods

	Name	Description
	AcceptLinks	Determines whether the node can be connected with links of the specified type. (Inherited from DiagramNode.)
	AddLabel	Overloaded. Adds a new label to this node. (Inherited from DiagramNode.)
	Animate	Animates the item. (Inherited from DiagramItem.)
	AttachManipulator	Adds the specified manipulator to this node. (Inherited from DiagramNode.)
	AttachTo	Overloaded. Attaches this node to another diagram item. (Inherited from DiagramNode.)
	Clone	Creates a copy of this item.
	Collapse	Hides all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	ContainsPoint	Overloaded. (Inherited from DiagramNode.)
	CreateConnectionPoint	Creates a connection point between this node and the specified link at the specified point. (Inherited from DiagramNode.)
	Detach	Detaches this node from its master item. (Inherited from DiagramNode.)
	DetachManipulator	Removes the specified manipulator from this node. (Inherited from DiagramNode.)
	Dispose	Overloaded. (Inherited from DiagramItem.)
	DoModify	For internal use. (Inherited from DiagramItem.)
	Draw	DiagramItem.Draw override. Draws the item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawAnchors	Draws the anchor points of this node. (Inherited from DiagramNode.)
	DrawHandles	DiagramItem.DrawHandles override. Renders the adjustment handles of this item. (Inherited from DiagramNode.)
	DrawInRect	(Inherited from DiagramNode.)
	DrawLocal	Override in a derived class to provide implementation specific drawing for the node. (Inherited from DiagramNode.)
	DrawShadow	DiagramItem.DrawShadow override. Draws the shadow of this item on the specified IGraphics surface. (Inherited from DiagramNode.)
	DrawShadowLocal	Override in a derived class to provide implementation specific drawing for the shadow of the node. (Inherited from DiagramNode.)

	Expand	Shows all nodes and links in the subtree starting from this node. (Inherited from DiagramNode.)
	GetAllIncomingLinks	Overloaded. (Inherited from DiagramNode.)
	GetAllLinks	Gets all links connected to this node. (Inherited from DiagramNode.)
	GetBounds	Gets the bounding rectangle of this item. (Inherited from DiagramItem.)
	GetCenter	Gets the center of this node's bounding rectangle. (Inherited from DiagramNode.)
	GetClosestAnchorPoint	Returns the index of the anchor point that lies at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetEffectiveBrush	Creates a Brush object representing the currently effective brush of the item. (Inherited from DiagramItem.)
	GetEffectivePen	Returns a Pen object representing the currently effective pen of the item. (Inherited from DiagramItem.)
	GetEffectiveShadowBrush	Returns a Brush object representing the currently effective shadow brush of the item. (Inherited from DiagramItem.)
	GetEffectiveTextBrush	Returns a Brush object representing the currently effective text brush of the item. (Inherited from DiagramItem.)
	GetIntersection	Calculates the intersection point between the segment defined by the specified point pair and the node's outline. (Inherited from DiagramNode.)
	GetNearestBorderPoint	Returns a point that lies on the outline of this node, at the shortest distance from the specified point. (Inherited from DiagramNode.)
	GetOutlinePoly	DiagramItem.GetOutlinePoly override. (Inherited from DiagramNode.)
	GetRepaintRect	DiagramItem.GetRepaintRect override. (Inherited from DiagramNode.)
	GetRotatedBounds	Overloaded. Gets the bounding rectangle of this node or the bounding rectangle of another rectangle rotated at a specified angle around a specified point. (Inherited from DiagramNode.)
	GetRotationAngle	Gets the rotation angle of this node. (Inherited from DiagramNode.)
	GetToolTip	DiagramItem.GetToolTip override. (Inherited from DiagramNode.)
	HitTest	Returns the diagram item that contains the specified point. (Inherited from DiagramItem.)
	HitTestHandle	DiagramItem.HitTestHandle override. Determines whether a selection handle of this item is located at the specified point. (Inherited from DiagramNode.)
	HitTestManipulators	DiagramItem.HitTestManipulators override. (Inherited from DiagramNode.)
	Intersects	Determines whether the node intersects another one, taking into account the outlines of node shapes. (Inherited from DiagramNode.)
	LabelFromPoint	Returns the label whose layout rectangle contains the specified point. (Inherited from DiagramNode.)
	Move	Moves the node to the specified location. (Inherited from DiagramNode.)
	NotInteractive	Gets whether the user is prevented from manipulating this item. (Inherited from DiagramItem.)
	NowModifying	Gets a flag indicating whether this object is being currently modified interactively by the user. (Inherited from DiagramItem.)
	OnDragOut	Invoked when a diagram item is being dragged out of this node. (Inherited from DiagramNode.)
	OnDragOver	Invoked when a diagram item is being dragged over this node. (Inherited from DiagramNode.)

	OnDropOver	Invoked when a diagram item is dropped over this node. (Inherited from DiagramNode.)
	OnLoad	Called after successful deserialization. (Inherited from DiagramItem.)
	Query	Returns the items that can be reached from this item by following the path specified in the given query expression. (Inherited from DiagramItem.)
	ReassignAnchorPoints	Reassigns links to anchor points so that they are evenly distributed. (Inherited from DiagramNode.)
	ReassignUnanchoredLinks	Reassigns unanchored links (Inherited from DiagramNode.)
	RemoveLabel	Removes the specified label from this node. (Inherited from DiagramNode.)
	Resize	Resizes the node. (Inherited from DiagramNode.)
	RestoreItemProperties	For internal use. (Inherited from DiagramItem.)
	SaveItemProperties	For internal use. (Inherited from DiagramItem.)
	SetBounds	Sets the location and size of this node. (Inherited from DiagramNode.)
	SetExpandedFlag	Sets the 'expanded' flag, repainting the node. (Inherited from DiagramNode.)
	SetPos	Sets the bounding rectangle of this node without updating connected arrows. (Inherited from DiagramNode.)
	SetRect	This method is used internally by the **Ruler** control to align nodes. (Inherited from DiagramNode.)
	UpdateAnimation	Invoked by running animations to update an item according to current animation progress. (Inherited from DiagramNode.)
	ZBottom	Moves the item to the bottom of the z-order. (Inherited from DiagramItem.)
	ZLevelDown	Moves the item one level down in the z-order. (Inherited from DiagramItem.)
	ZLevelUp	Moves the item one level up in the z-order. (Inherited from DiagramItem.)
	ZTop	Moves the item to the top of the z-order. (Inherited from DiagramItem.)

Protected Methods

	Name	Description
	AddManipulator	Adds the specified manipulator to this object. (Inherited from DiagramItem.)
	AllowCreate	DiagramItem.AllowCreate override. (Inherited from DiagramNode.)
	AllowModify	DiagramItem.AllowModify override. (Inherited from DiagramNode.)
	CancelModify	DiagramItem.CancelModify override. Invoked when the interactive modification of this object has been canceled. (Inherited from DiagramNode.)
	CompleteCreate	DiagramItem.CompleteCreate override. (Inherited from DiagramNode.)
	CompleteModify	DiagramItem.CompleteModify override. (Inherited from DiagramNode.)
	ConstraintsInfringed	Check whether the KeepInsideParent constraint is being infringed. (Inherited from DiagramNode.)
	CreateProperties	DiagramItem.CreateProperties override. (Inherited from DiagramNode.)
	CreateState	DiagramItem.CreateState override. (Inherited from DiagramNode.)

	DrawManipulators	Overloaded. Draws the associated manipulators on a specified graphics. (Inherited from DiagramItem.)
	Finalize	DiagramItem finalizer. (Inherited from DiagramItem.)
	FreeResources	Called when the node is completely disassociated from a Diagram, that is, when it is no longer referenced from an item collection or the undo/redo history. (Inherited from DiagramNode.)
	GetAllOutgoingLinks	Overloaded. (Inherited from DiagramNode.)
	GetLocalBounds	Gets this node's bounding rectangle in local coordinates. (Inherited from DiagramNode.)
	GetRgnInRect	Creates and returns a **Region** object for this node as though it is contained in the specified rectangle. (Inherited from DiagramNode.)
	GetValue	Gets the effective value of the style property with the specified name. (Inherited from DiagramItem.)
	GroupContains	Determines whether this node is a predecessor of the specified node in the group hierarchy. (Inherited from DiagramNode.)
	LoadFrom	Deserializes the object from the specified reader.
	LoadFromJson	Loads the item content from a JsonObject.
	LoadFromXml	Loads the item content from an XML element.
	LoadObsolete	For internal use. (Inherited from DiagramItem.)
	OnAdd	DiagramItem.OnAdd override. (Inherited from DiagramNode.)
	OnChildModified	Called when a group child node has been modified. (Inherited from DiagramItem.)
	OnClick	Invoked when the user clicks this item. (Inherited from DiagramItem.)
	OnDoubleClick	Invoked when the user double-clicks this item. (Inherited from DiagramItem.)
	OnMeasureUnitChanged	DiagramItem.OnMeasureUnitChanged override. (Inherited from DiagramNode.)
	OnModify	Invoked when the user moves or resizes this node. (Inherited from DiagramNode.)
	OnRemove	DiagramItem.OnRemove override. (Inherited from DiagramNode.)
	OnRotationAngleChanged	Invoked when the rotation angle of this node has been changed, either programmatically or interactively. (Inherited from DiagramNode.)
	OnStyleChanged	Invoked when the associated style or theme have changed. (Inherited from DiagramItem.)
	OnTextChanged	Called when the value of the Text property changes. (Inherited from DiagramItem.)
	OnTextFormatChanged	Called when the value of the TextFormat property changes. (Inherited from DiagramItem.)
	OnUpdateBounds	Overloaded. (Inherited from DiagramNode.)
	OnZoomChanged	DiagramItem.OnZoomChanged override. (Inherited from DiagramNode.)
	RemoveManipulator	Removes the specified manipulator from the object. (Inherited from DiagramItem.)
	Repaint	Overloaded. (Inherited from DiagramItem.)
	RestoreProperties	DiagramItem.RestoreProperties override. (Inherited from DiagramNode.)
	RestoreState	DiagramItem.RestoreState override. (Inherited from DiagramNode.)

	SaveProperties	DiagramItem.SaveProperties override. (Inherited from DiagramNode.)
	SaveState	DiagramItem.SaveState override. (Inherited from DiagramNode.)
	SaveTo	Serializes the object to the specified writer.
	SaveToJson	Saves the item content into a JsonObject.
	SaveToXml	Saves the item content into an XML element.
	ScaleElements	Scales the node's elements by specified amount. (Inherited from DiagramNode.)
	SetDiagramDirty	Marks the diagram as changed. (Inherited from DiagramItem.)
	SetExpandable	Internally sets the expandable state of this object to the specified value, adding or removing an Expander manipulator object if necessary. (Inherited from DiagramNode.)
	SetExpanded	DiagramItem.SetExpanded override. (Inherited from DiagramNode.)
	SetIncomingLinks	Sets the incoming links collection of this node. (Inherited from DiagramNode.)
	SetOutgoingLinks	Sets the outgoing links collection of this node. (Inherited from DiagramNode.)
	SetReflexive	Invoked to let a node set the shape of a reflexive link connected to it. (Inherited from DiagramNode.)
	ShouldRender	Determines whether the item should be rendered in the current context. (Inherited from DiagramItem.)
	ShouldRenderAnchors	Determines whether the node should render its anchor points. (Inherited from DiagramNode.)
	StartCreate	DiagramItem.StartCreate override. (Inherited from DiagramNode.)
	StartModify	DiagramItem.StartModify override. Invoked when the user starts modifying this item interactively. (Inherited from DiagramNode.)
	UpdateCreate	DiagramItem.UpdateCreate override. (Inherited from DiagramNode.)
	UpdateGroup	Called to update the positions of subordinate nodes attached to this item when the attachment style is set to ControlledByMaster. (Inherited from DiagramItem.)
	UpdateLinkIntersections	Updates the intersection points of links, connected to this node and its outline, usually in response to change in the node's shape. (Inherited from DiagramNode.)
	UpdateLinkPositions	Updates the end points of the links, connected to this node, usually in response to change in the node's size or shape. (Inherited from DiagramNode.)
	UpdateModify	DiagramItem.UpdateModify override. Invoked during interactive modification of this item. (Inherited from DiagramNode.)

See Also

ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode Constructor

Overload List

Name	Description

ControlNode ()	Initializes a new instance of the ControlNode class.
ControlNode (ControlNode)	Initializes a new instance of the ControlNode class.
ControlNode (Diagram)	Initializes a new instance of the ControlNode class.

See Also

ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode Constructor ()

Initializes a new instance of the ControlNode class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ControlNode ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ControlNode Constructor Overload List
ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode Constructor (ControlNode)

Initializes a new instance of the ControlNode class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ControlNode (
    ControlNode prototype
)
```

Visual Basic Copy Code

```
Public New ( _
    prototype As ControlNode _
)
```

Parameters

prototype

See Also

ControlNode Constructor Overload List
ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode Constructor (Diagram)

Initializes a new instance of the ControlNode class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ControlNode (
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public New ( _
    diagram As Diagram _
)
```

Parameters

diagram

See Also

ControlNode Constructor Overload List
ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode Methods

ControlNode.Clone Method

Creates a copy of this item.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override DiagramItem Clone (
    bool clipboard
)
```

Visual Basic Copy Code

```
Public Overrides Function Clone( _
    clipboard As Boolean _
) As DiagramItem
```

Parameters

clipboard

Return Value

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode.LoadFrom Method

Deserializes the object from the specified reader.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
protected override void LoadFrom (
    BinaryReader reader,
    PersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFrom( _
    reader As BinaryReader, _
    context As PersistContext _
)
```

Parameters

reader
A BinaryReader object from which to read the item's data.

context
A PersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode.LoadFromJson Method

Loads the item content from a JsonObject.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
protected override void LoadFromJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject containing the item's data.

context
A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode.LoadFromXml Method

Loads the item content from an XML element.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
protected override void LoadFromXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadFromXml( _
    xmlElement As XmlElement, _
    context As XmlPersistContext _
)
```

Parameters

xmlElement

context

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode.SaveTo Method

Serializes the object to the specified writer.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
protected override void SaveTo (
    BinaryWriter writer,
    PersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveTo( _
    writer As BinaryWriter, _
    context As PersistContext _
)
```

Parameters

writer
A BinaryWriter object the item's data should be written to.

context
A PersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode.SaveToJson Method

Saves the item content into a JsonObject.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
protected override void SaveToJson (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToJson( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj
A JsonObject the item's data should be stored into.

context
A JsonPersistContext object providing contextual information about the serialization process and some helper serialization methods.

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode.SaveToXml Method

Saves the item content into an XML element.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
protected override void SaveToXml (
    XmlElement xmlElement,
    XmlPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveToXml( _
    xmlElement As XmlElement, _
```

```
    context As XmlPersistContext _
)
```

Parameters

xmlElement

context

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNode Properties

ControlNode.Template Property

Gets or sets the HTML string representing the DOM content rendered in this node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Template { get; set; }
```

Visual Basic Copy Code

```
Public Property Template As String
```

Property Value

See Also

ControlNode Members
ControlNode Class
MindFusion.Diagramming.Mvc Namespace

ControlNodeConverter Class

Serializes ControlNode instances.

Syntax

C# Copy Code

```
public class ControlNodeConverter : DiagramNodeConverter
```

Visual Basic Copy Code

```
Public Class ControlNodeConverter
    Inherits DiagramNodeConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.DiagramItemConverter
      MindFusion.Diagramming.Mvc.DiagramNodeConverter
        MindFusion.Diagramming.Mvc.ControlNodeConverter

See Also

ControlNodeConverter Members
MindFusion.Diagramming.Mvc Namespace

ControlNodeConverter Members

Serializes ControlNode instances.

The following tables list the members exposed by the ControlNodeConverter type.

Public Constructors

	Name	Description
	ControlNodeConverter	Creates a new instance of the ControlNodeConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

ControlNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ControlNodeConverter Constructor

Creates a new instance of the ControlNodeConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ControlNodeConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

ControlNodeConverter Methods

ControlNodeConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

ControlNodeConverter Members
ControlNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ControlNodeConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
```

```
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

ControlNodeConverter Members
ControlNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ControlNodeConverter Properties

ControlNodeConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

ControlNodeConverter Members
ControlNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ControlNodeStyle Class

Represents a reusable set of properties that define the appearance of control nodes.

Syntax

C# Copy Code

```
public class ControlNodeStyle : DiagramNodeStyle
```

Visual Basic Copy Code

```
Public Class ControlNodeStyle
    Inherits DiagramNodeStyle
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Style
    MindFusion.Diagramming.DiagramItemStyle

MindFusion.Diagramming.DiagramNodeStyle
MindFusion.Diagramming.Mvc.ControlNodeStyle

See Also

ControlNodeStyle Members
MindFusion.Diagramming.Mvc Namespace

ControlNodeStyle Members

Represents a reusable set of properties that define the appearance of control nodes.

The following tables list the members exposed by the ControlNodeStyle type.

Public Constructors

	Name	Description
	ControlNodeStyle	Initializes a new instance of the ControlNodeStyle class.

Public Properties

	Name	Description
	Brush	Gets or sets the brush used to fill the interior of the item. (Inherited from DiagramItemStyle.)
	Effects	Gets a list of the effects applied to the node. (Inherited from DiagramNodeStyle.)
	FontFamily	Gets or sets the font family of the item text. (Inherited from DiagramItemStyle.)
	FontSize	Gets or sets the font size of the item text. (Inherited from DiagramItemStyle.)
	FontStyle	Gets or sets the font style of the item text. (Inherited from DiagramItemStyle.)
	FontUnit	Gets or sets the font unit of the item text. (Inherited from DiagramItemStyle.)
	ShadowBrush	Gets or sets the brush used to paint the shadow of the item. (Inherited from DiagramItemStyle.)
	Stroke	Gets or sets the brush used to paint the outline of the item. (Inherited from DiagramItemStyle.)
	StrokeDashStyle	Gets or sets the pattern of dashes and gaps used to outline items. (Inherited from DiagramItemStyle.)
	StrokeThickness	Gets or sets the thickness of the item's outline, expressed in the current units of measure. (Inherited from DiagramItemStyle.)
	TextBrush	Gets or sets the brush used to paint texts inside the item. (Inherited from DiagramItemStyle.)

Public Methods

	Name	Description
	GetProperties	Gets all properties in the style. (Inherited from Style.)
	GetProperty	Returns the property with the specified name, if any. (Inherited from Style.)
S	RegisterType	Associates a type with an alias. (Inherited from Style.)

Protected Methods

	Name	Description
	GetValue	Gets the value of the specified property. (Inherited from Style.)
	IsPropertySet	Checks whether the specified property has a value. (Inherited from Style.)
	LoadFromJson	Loads the style content from a JsonObject. (Inherited from DiagramNodeStyle.)
	LoadFromXml	Loads the style content from an XML element. (Inherited from Style.)
	OnPropertyChanged	Raises the PropertyChanged event. (Inherited from Style.)
	RegisterProperty	Registers a new property with the specified name and type. (Inherited from Style.)
	SaveToJson	Saves the style content into a JsonObject. (Inherited from DiagramNodeStyle.)
	SaveToXml	Saves the style content into an XML element. (Inherited from Style.)
	SetValue	Sets the value of the specified property. (Inherited from Style.)
	UnsetValue	Clears the value of the specified property. (Inherited from Style.)

Public Events

	Name	Description
	PropertyChanged	Raised when any of the registered style properties has changed. (Inherited from Style.)

See Also

ControlNodeStyle Class
MindFusion.Diagramming.Mvc Namespace

ControlNodeStyle Constructor

Initializes a new instance of the ControlNodeStyle class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ControlNodeStyle ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ControlNodeStyle Class
MindFusion.Diagramming.Mvc Namespace

ControlNodeStyle Events

DiagramItemConverter Class

Serializes DiagramItem instances.

Syntax

C# Copy Code

```
public class DiagramItemConverter : JavaScriptConverter
```

Visual Basic Copy Code

```
Public Class DiagramItemConverter
    Inherits JavaScriptConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.DiagramItemConverter
      MindFusion.Diagramming.Mvc.DiagramLinkConverter
      MindFusion.Diagramming.Mvc.DiagramNodeConverter

See Also

DiagramItemConverter Members
MindFusion.Diagramming.Mvc Namespace

DiagramItemConverter Members

Serializes DiagramItem instances.

The following tables list the members exposed by the DiagramItemConverter type.

Public Constructors

	Name	Description
	DiagramItemConverter	Creates a new instance of the DiagramItemConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

DiagramItemConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramItemConverter Constructor

Creates a new instance of the DiagramItemConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramItemConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

DiagramItemConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramItemConverter Methods

DiagramItemConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

DiagramItemConverter Members
DiagramItemConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramItemConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

DiagramItemConverter Members
DiagramItemConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramItemConverter Properties

DiagramItemConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

DiagramItemConverter Members
DiagramItemConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramLinkConverter Class

Serializes DiagramLink instances.

Syntax

C# Copy Code

```
public class DiagramLinkConverter : DiagramItemConverter
```

Visual Basic Copy Code

```
Public Class DiagramLinkConverter
    Inherits DiagramItemConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.DiagramItemConverter
      MindFusion.Diagramming.Mvc.DiagramLinkConverter

See Also

DiagramLinkConverter Members
MindFusion.Diagramming.Mvc Namespace

DiagramLinkConverter Members

Serializes DiagramLink instances.

The following tables list the members exposed by the DiagramLinkConverter type.

Public Constructors

	Name	Description
	DiagramLinkConverter	Creates a new instance of the DiagramLinkConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

DiagramLinkConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramLinkConverter Constructor

Creates a new instance of the DiagramLinkConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramLinkConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

DiagramLinkConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramLinkConverter Methods

DiagramLinkConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

DiagramLinkConverter Members
DiagramLinkConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramLinkConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

DiagramLinkConverter Members
DiagramLinkConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramLinkConverter Properties

DiagramLinkConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

DiagramLinkConverter Members
DiagramLinkConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramNodeConverter Class

Serializes DiagramNode instances.

Syntax

C# Copy Code

```
public class DiagramNodeConverter : DiagramItemConverter
```

Visual Basic Copy Code

```
Public Class DiagramNodeConverter
    Inherits DiagramItemConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.DiagramItemConverter
      MindFusion.Diagramming.Mvc.DiagramNodeConverter
        MindFusion.Diagramming.Mvc.CompositeNodeConverter
        MindFusion.Diagramming.Mvc.ContainerNodeConverter
        MindFusion.Diagramming.Mvc.ControlNodeConverter
        MindFusion.Diagramming.Mvc.FreeFormNodeConverter
        MindFusion.Diagramming.Mvc.ShapeNodeConverter
        MindFusion.Diagramming.Mvc.SvgNodeConverter
        MindFusion.Diagramming.Mvc.TableNodeConverter
        MindFusion.Diagramming.Mvc.TreeViewNodeConverter
        MindFusion.Diagramming.Mvc.VideoNodeConverter

See Also

DiagramNodeConverter Members
MindFusion.Diagramming.Mvc Namespace

DiagramNodeConverter Members

Serializes DiagramNode instances.

The following tables list the members exposed by the DiagramNodeConverter type.

Public Constructors

	Name	Description
	DiagramNodeConverter	Creates a new instance of the DiagramNodeConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

DiagramNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramNodeConverter Constructor

Creates a new instance of the DiagramNodeConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramNodeConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

DiagramNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramNodeConverter Methods

DiagramNodeConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

DiagramNodeConverter Members
DiagramNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramNodeConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

DiagramNodeConverter Members
DiagramNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramNodeConverter Properties

DiagramNodeConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

DiagramNodeConverter Members
DiagramNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramView Class

A control that displays the contents of a Diagram on a web page.

Syntax

C# Copy Code

```
public class DiagramView
```

Visual Basic Copy Code

```
Public Class DiagramView
```

Remarks

Apart from rendering the diagram, **DiagramView** also controls most aspects of user interaction. The Behavior property specifies how the control responds to user actions with the mouse. AllowInplaceEdit specifies whether users are allowed to edit in-place the content of a diagram element by double-clicking it.

DiagramView can be created inside a Razor page either by invoking the DiagramView html helper, or by adding a <mindfusion-diagramview> tag helper:

Razor Copy Code

```
@addTagHelper *, MindFusion.Diagramming.Mvc
....
<mindfusion-diagramview
    component=(DiagramView)ViewBag.DiagramView
    style="width:100%; height:100%;">
</mindfusion-diagramview>
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.DiagramView
    MindFusion.Diagramming.Mvc.TabbedDiagramView

See Also

DiagramView Members
MindFusion.Diagramming.Mvc Namespace

DiagramView Members

A control that displays the contents of a Diagram on a web page.

The following tables list the members exposed by the DiagramView type.

Public Constructors

	Name	Description
	DiagramView	Overloaded. Creates an instance of **DiagramView**.

Public Properties

	Name	Description

	AddScriptReferences	Gets or sets a value indicating whether the control should render references to the client scripts.
	AllowInplaceEdit	Gets or sets a value indicating whether in-place editing of the text of objects is enabled.
	AutoClasses	Gets the registered client types with JSON templates.
	AutoScroll	Gets or sets a value indicating whether auto scrolling of the document area is enabled.
	Behavior	Gets or sets a value indicating how the control responds to user actions.
	ClientScriptLocation	Gets or sets the URL to the MindFusion.Diagramming.js file.
	Converters	Gets the registered custom converters.
	CustomLinkType	Gets the type of a MindFusion.Diagramming.DiagramLink derived class whose instance should be created when a user starts drawing.
	CustomNodeType	Gets the type of a MindFusion.Diagramming.DiagramNode derived class whose instance should be created when a user starts drawing.
	DefaultControlTemplate	Gets or sets the HTML string representing the DOM content, rendered by default in newly created ControlNode objects.
	DelKeyAction	Gets or sets a value indicating the action that should be performed when the user hits the DELETE key.
	Diagram	Gets or sets the **Diagram** object displayed by this view.
	Enabled	Gets or sets a value indicating whether handling of mouse events is enabled.
	EnableMultiTouch	Gets or sets a value indicating whether the control handles DOM pointer events to implement multi-touch interactions, such as zooming, node rotation or simultaneous drawing of multiple diagram items.
	FallbackContent	Gets or sets the HTML to be rendered as the canvas' fallback content.
	HandleTouchHitDistance	Gets or sets maximum distance between touch position and adjustment handle at which the handle can still be grabbed to modify its item.
	Height	Gets or sets the height of the view.
	Id	Gets or sets the id of the view.
	ItemTypes	Gets the registered item types.
	LicenseKey	Gets or sets the license key of the control.
	MagnifierEnabled	Gets or sets a value indicating whether the magnifier tool is currently enabled.
	MagnifierFactor	Gets or sets the zoom factor of the magnifier tool.
	MagnifierFrameColor	Gets or sets the color of the magnifier frame.
	MagnifierFrameThickness	Gets or sets the thickness of the magnifier frame, expressed in pixels.
	MagnifierHeight	Gets or sets the height of the magnifier tool, expressed in pixels.
	MagnifierSecondaryFrameColor	Gets or sets the secondary color of the magnifier frame.
	MagnifierShading	Gets or sets a value indicating whether to enhance the effect of a magnifier lense by using a gradient shading.
	MagnifierShape	Gets or sets the shape of the magnifier tool.
	MagnifierWidth	Gets or sets the width of the magnifier tool, expressed in pixels.
	MaxJsonLength	Gets or sets the maximum length of JSON strings that are accepted by internal JavaScriptSerializer instances.

	MinVisibleFontSize	Gets or sets a threshold value that hides text if scaled font sizes become smaller.
	ModificationStart	Gets or sets how users can start modifying diagram items.
	MouseWheelAction	Gets or sets what action to perform when users rotate the mouse wheel.
	MultiTouchDraw	Gets or sets a value indicating whether the control allows drawing multiple items simultaneously.
	MultiTouchModify	Gets or sets a value indicating whether multi-touch modifications are enabled.
	MultiTouchZoom	Gets or sets a value indicating whether multi-touch zoom is enabled.
	RecursionLimit	Gets or sets the limit for constraining the number of object levels to process by internal JavaScriptSerializer instances.
	ScrollLeft	Gets or sets the view's X scroll position.
	ScrollTop	Gets or sets the view's Y scroll position.
	ShapeLibraryLocation	The location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path.
	TooltipDelay	Gets or sets the delay before a tooltip is shown.
	VirtualScroll	Gets or sets a value indicating whether virtual scroll mode is enabled.
	Width	Gets or sets the width of the view.
	ZoomFactor	Gets or sets the zoom of the view.

Public Methods

	Name	Description
	AnimatedLayoutCompletedScript	Sets the name of a JavaScript function that should be invoked when an animated layout completes.
	CellTextEditedScript	Sets the name of a JavaScript function that should be invoked when the user edits the text of a table cell.
	CellTextEditingScript	Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a table cell.
	ClickedScript	Sets the name of a JavaScript function that should be invoked when the user clicks the diagram at a point where no items are located.
	ContainerChildAddedScript	Sets the name of a JavaScript function that should be invoked when a child node is added to a container.
	ContainerChildAddingScript	Sets the name of a JavaScript function that should be invoked before a child node is added to a container.
	ContainerChildRemovedScript	Sets the name of a JavaScript function that should be invoked when a child node is removed from a container.
	ContainerChildRemovingScript	Sets the name of a JavaScript function that should be invoked before a child node is removed from a container.
	ContainerFoldedScript	Sets the name of a JavaScript function that should be invoked when the user folds a container by clicking the arrow-up button in the container's caption area.
	ContainerUnfoldedScript	Sets the name of a JavaScript function that should be invoked when the user unfolds a container by clicking the arrow-down button in the container's caption area.
	ControlLoadedScript	Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.

	CreateEditControlScript	Sets the name of a JavaScript function that should be invoked when when the inplace-edit control is created.
	DataLoadedScript	Sets the name of a JavaScript function that should be invoked when the diagram data is loaded.
	DeserializeTagScript	Sets the name of a JavaScript function that should be invoked when the Tag or Id of a diagram item must be deserialized from XML format.
	DiagramToView	Overloaded. Transforms points from the document coordinate space to the client coordinate space.
	DrawAdjustmentHandlesScript	Sets the name of a JavaScript function that should be invoked when item's adjustment handles must be custom drawn.
	EnterInplaceEditScript	Sets the name of a JavaScript function that should be invoked when when entering inplace-edit mode.
S	FromJson	Overloaded.
S	FromRequest	Overloaded.
	HeaderResizedScript	Sets the name of a JavaScript function that should be invoked when the user finished resizing a header.
	HeaderStartResizingScript	Sets the name of a JavaScript function that should be invoked when the user starts resizing a header.
	HitTestAdjustmentHandlesScript	Sets the name of a JavaScript function that should be invoked when custom hit testing of item's adjustment handles must be performed.
	HyperlinkClickedScript	Sets the name of a JavaScript function that should be invoked when the user clicks a hyperlink inside a node.
	Init	Renders the DiagramView's initializing scripts on a web page under the specified function name.
	InitializeLinkScript	Sets the name of a JavaScript function that should be invoked when the user starts drawing a new link, just after the DiagramLink instance is created.
	InitializeNodeScript	Sets the name of a JavaScript function that should be invoked when the user starts drawing a new node, just after the DiagramNode instance is created.
	ItemAddedScript	Sets the name of a JavaScript function that should be invoked when an item is added to the diagram.
	ItemRemovedScript	Sets the name of a JavaScript function that should be invoked when an item is removed from the diagram.
	LaneGridCellTextEditedScript	Sets the name of a JavaScript function that should be invoked hen the used edits the text of a grid cell.
	LeaveInplaceEditScript	Sets the name of a JavaScript function that should be invoked when when leaving inplace-edit mode.
	LinkClickedScript	Sets the name of a JavaScript function that should be invoked when the users clicks a link.
	LinkCreatedScript	Sets the name of a JavaScript function that should be invoked when the user draws a new link.
	LinkCreatingScript	Sets the name of a JavaScript function that should be invoked while the user is drawing a new link.
	LinkDeletedScript	Sets the name of a JavaScript function that should be invoked when a link is deleted, either programmatically or by the user.
	LinkDeletingScript	Sets the name of a JavaScript function that should be invoked when the user tries to delete a link.
	LinkDeselectedScript	Sets the name of a JavaScript function that should be invoked when a link is deselected, either programmatically or by the user.

	LinkDoubleClickedScript	Sets the name of a JavaScript function that should be invoked when the users double-clicks a link.
	LinkModifiedScript	Sets the name of a JavaScript function that should be invoked when the user moves any control point of a link.
	LinkModifyingScript	Sets the name of a JavaScript function that should be invoked while the user is modifying a link.
	LinkPastedScript	Sets the name of a JavaScript function that should be invoked when a diagram link is pasted from clipboard.
	LinkPointedScript	Sets the name of a JavaScript function that should be invoked when the mouse pointer hovers over a link.
	LinkSelectedScript	Sets the name of a JavaScript function that should be invoked when a link is selected, either programmatically or by the user.
	LinkSelectingScript	Sets the name of a JavaScript function that should be invoked when the user tries to select a link.
	LinkTextEditedScript	Sets the name of a JavaScript function that should be invoked when the user edits the text of a link.
	LinkTextEditingScript	Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a link.
	LoadFromJson	Overloaded.
	MouseWheelScript	Sets the name of a JavaScript function that should be invoked when the user rotates the mouse wheel while the cursor is positioned over the diagram surface.
	NodeClickedScript	Sets the name of a JavaScript function that should be invoked when the users clicks a node.
	NodeCreatedScript	Sets the name of a JavaScript function that should be invoked when the user draws a new node.
	NodeCreatingScript	Sets the name of a JavaScript function that should be invoked while the user is drawing a new node.
	NodeDeletedScript	Sets the name of a JavaScript function that should be invoked when a node is deleted, either programmatically or by the user.
	NodeDeletingScript	Sets the name of a JavaScript function that should be invoked when the user tries to delete a node.
	NodeDeselectedScript	Sets the name of a JavaScript function that should be invoked when a node is deselected, either programmatically or by the user.
	NodeDomCreatedScript	Sets the name of a JavaScript function that should be invoked when the content of a ControlNode is added to the page DOM.
	NodeDoubleClickedScript	Sets the name of a JavaScript function that should be invoked when the users double-clicks a node.
	NodeModifiedScript	Sets the name of a JavaScript function that should be invoked when the user moves or resizes a node.
	NodeModifyingScript	Sets the name of a JavaScript function that should be invoked while the user is moving or resizing a node.
	NodePastedScript	Sets the name of a JavaScript function that should be invoked when a diagram node is pasted from clipboard.
	NodePointedScript	Sets the name of a JavaScript function that should be invoked when the mouse pointer hovers over a node.
	NodeSelectedScript	Sets the name of a JavaScript function that should be invoked when a node is selected, either programmatically or by the user.
	NodeSelectingScript	Sets the name of a JavaScript function that should be invoked when the user tries to select a node.

	NodeTextEditedScript	Sets the name of a JavaScript function that should be invoked when the user edits the text of a node.
	NodeTextEditingScript	Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a node.
	RegisterConverters	Registers custom converters used for custom item types serialization.
	RegisterItemType	Overloaded.
	RegisterSession	Overloaded.
	Render	Renders the **DiagramView** on a web page.
	RepaintScript	Sets the name of a JavaScript function that should be invoked when the diagram has been repainted.
	SaveToJson	Serializes the DiagramView's data to JSON.
	SelectionChangedScript	Sets the name of a JavaScript function that should be invoked when the diagram selection has changed.
	SelectionModifyingScript	Sets the name of a JavaScript function that should be invoked while the user is moving or resizing multiple items.
	SerializeTagScript	Sets the name of a JavaScript function that should be invoked when the Tag or Id of a diagram item must be serialized in XML format.
	SetAddScriptReferences	Sets a value indicating whether the control should render references to the client scripts.
	SetAllowInplaceEdit	Sets a value indicating whether in-place editing of the text of objects is enabled.
	SetAutoScroll	Sets a value indicating whether auto scrolling of the document area is enabled.
	SetBehavior	Sets a value indicating how the control responds to user actions.
	SetCustomLinkType	Sets the type of a MindFusion.Diagramming.**DiagramLink** derived class whose instance should be created when a user starts drawing.
	SetCustomNodeType	Sets the type of a MindFusion.Diagramming.**DiagramNode** derived class whose instance should be created when a user starts drawing.
	SetDelKeyAction	Sets a value indicating the action that should be performed when the user hits the Del key.
	SetEnabled	Sets a value indicating whether handling of mouse events is enabled.
	SetEnableMultiTouch	Sets a value indicating whethr the control handles DOM pointer events to implement multi-touch interactions, such as zooming, node rotation or simultaneous drawing of multiple diagram items.
	SetFallbackContent	Sets the HTML to be rendered as the canvas' fallback content.
	SetHeight	Sets the height of the view.
	SetMagnifierEnabled	Sets a value indicating whether the magnifier tool is currently enabled.
	SetMagnifierFactor	Sets the zoom factor of the magnifier tool.
	SetMagnifierFrameColor	Sets the color of the magnifier frame.
	SetMagnifierFrameThickness	Sets the thickness of the magnifier frame, expressed in pixels.
	SetMagnifierHeight	Sets the height of the magnifier tool, expressed in pixels.
	SetMagnifierSecondaryFrameColor	Sets the secondary color of the magnifier frame.
	SetMagnifierShading	Sets a value indicating whether to enhance the effect of a magnifier lense by using a gradient shading.
	SetMagnifierShape	Sets the shape of the magnifier tool.

	SetMagnifierWidth	Sets the width of the magnifier tool, expressed in pixels.
	SetMaxJsonLength	Sets the maximum length of JSON strings that are accepted by internal JavaScriptSerializer instances.
	SetMinVisibleFontSize	Sets a threshold value that hides text if scaled font sizes become smaller.
	SetModificationStart	Sets how users can start modifying diagram items.
	SetMouseWheelAction	Sets a value indicating what action to perform when users rotate the mouse wheel.
	SetMultiTouchDraw	Sets a value indicating whether the control allows drawing multiple items simultaneously.
	SetMultiTouchModify	Sets a value indicating whether multi-touch modifications are enabled.
	SetMultiTouchZoom	Sets a value indicating whether multi-touch zoom is enabled.
	SetRecursionLimit	Sets the limit for constraining the number of object levels to process by internal JavaScriptSerializer instances.
	SetScrollLeft	Sets the view's X scroll position.
	SetScrollTop	Sets the view's Y scroll position.
	SetShowScrollbars	Sets a value indicating whether the scrollbars are visible.
	SetTooltipDelay	Sets the delay before a tooltip is shown.
	SetVirtualScroll	Sets a value indicating whether virtual scroll mode is enabled.
	SetWidth	Sets the width of the view.
	SetZoomFactor	Sets the zoom of the view.
	SizeChangedScript	Sets the name of a JavaScript function that should be invoked when the size of the diagram canvas has changed.
	TableColumnResizedScript	Sets the name of a JavaScript function that should be invoked when the user finishes resizing a column in a table interactively.
	TableColumnResizingScript	Sets the name of a JavaScript function that should be invoked when the user resizes a column in a table interactively.
	TableRowResizedScript	Sets the name of a JavaScript function that should be invoked when the user finishes resizing a row in a table interactively.
	TableRowResizingScript	Sets the name of a JavaScript function that should be invoked when the user resizes a row in a table interactively.
	ToJson	Serializes the DiagramView's data to JSON.
	TreeCollapsedScript	Sets the name of a JavaScript function that should be invoked when a user collapses a tree branch by clicking the [+] button of an Expandable node.
	TreeExpandedScript	Sets the name of a JavaScript function that should be invoked when a user expands a tree branch by clicking the [+] button of an Expandable node.
	TreeItemTextEditedScript	Sets the name of a JavaScript function that should be invoked when the user edits the text of a tree view item.
	TreeItemTextEditingScript	Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a tree view item.
	ViewToDiagram	Overloaded.

Protected Methods

	Name	Description

	GetAttributes	
	LoadViewProps	
	SaveViewProps	

See Also

DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView Constructor

Creates an instance of DiagramView.

Overload List

Name	Description
DiagramView ()	Creates an instance of **DiagramView**.
DiagramView (String)	Creates an instance of **DiagramView**.
DiagramView (String, Diagram)	Creates an instance of **DiagramView**.

See Also

DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView Constructor ()

Creates an instance of DiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramView Constructor Overload List
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView Constructor (String)

Creates an instance of DiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public DiagramView (
    string id
)
```

Visual Basic

```
Public New ( _
    id As String _
)
```

Parameters

id

The Id of the DiagramView.

See Also

DiagramView Constructor Overload List
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView Constructor (String, Diagram)

Creates an instance of DiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public DiagramView (
    string id,
    Diagram diagram
)
```

Visual Basic

```
Public New ( _
    id As String, _
    diagram As Diagram _
)
```

Parameters

id

The Id of the DiagramView.

diagram

The Diagram displayed by the view.

See Also

DiagramView Constructor Overload List
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView Methods

DiagramView.AnimatedLayoutCompletedScript Method

Sets the name of a JavaScript function that should be invoked when an animated layout completes.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView AnimatedLayoutCompletedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function AnimatedLayoutCompletedScript( _
    value As String _
) As DiagramView
```

Parameters

value
Name of the animatedLayoutCompleted event handler function.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.CellTextEditedScript Method

Sets the name of a JavaScript function that should be invoked when the user edits the text of a table cell.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView CellTextEditedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function CellTextEditedScript( _
    value As String _
) As DiagramView
```

Parameters

value
Name of the cellTextEdited event handler function.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.CellTextEditingScript Method

Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a table cell.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView CellTextEditingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function CellTextEditingScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ClickedScript Method

Sets the name of a JavaScript function that should be invoked when the user clicks the diagram at a point where no items are located.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView ClickedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ClickedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ContainerChildAddedScript Method

Sets the name of a JavaScript function that should be invoked when a child node is added to a container.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView ContainerChildAddedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ContainerChildAddedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ContainerChildAddingScript Method

Sets the name of a JavaScript function that should be invoked before a child node is added to a container.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView ContainerChildAddingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ContainerChildAddingScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ContainerChildRemovedScript Method

Sets the name of a JavaScript function that should be invoked when a child node is removed from a container.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView ContainerChildRemovedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ContainerChildRemovedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ContainerChildRemovingScript Method

Sets the name of a JavaScript function that should be invoked before a child node is removed from a container.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView ContainerChildRemovingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ContainerChildRemovingScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ContainerFoldedScript Method

Sets the name of a JavaScript function that should be invoked when the user folds a container by clicking the arrow-up button in the container's caption area.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView ContainerFoldedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ContainerFoldedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ContainerUnfoldedScript Method

Sets the name of a JavaScript function that should be invoked when the user unfolds a container by clicking the arrow-down button in the container's caption area.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView ContainerUnfoldedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ContainerUnfoldedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView.ControlLoadedScript Method

Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView ControlLoadedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ControlLoadedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.CreateEditControlScript Method

Sets the name of a JavaScript function that should be invoked when when the inplace-edit control is created.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView CreateEditControlScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function CreateEditControlScript( _
    value As String _
) As DiagramView
```

Parameters

value
Name of the createEditControl event handler function.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.DataLoadedScript Method

Sets the name of a JavaScript function that should be invoked when the diagram data is loaded.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView DataLoadedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function DataLoadedScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.DeserializeTagScript Method

Sets the name of a JavaScript function that should be invoked when the Tag or Id of a diagram item must be deserialized from XML format.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView DeserializeTagScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function DeserializeTagScript( _
    value As String _
) As DiagramView
```

Parameters

value
Name of the deserializeTag event handler function.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.DiagramToView Method

Transforms points from the document coordinate space to the client coordinate space.

Overload List

Name	Description
DiagramView.DiagramToView (Point)	Converts the specified point from document to client coordinates.
DiagramView.DiagramToView (Rect)	Converts the specified rectangle from document to client coordinates.
DiagramView.DiagramToView (Single)	Converts the specified value from document to client coordinates.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.DiagramToView Method (Point)

Converts the specified point from document to client coordinates.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Point DiagramToView (
    Point docPoint
)
```

Visual Basic Copy Code

```
Public Function DiagramToView( _
    docPoint As Point _
) As Point
```

Parameters

docPoint
The **Microsoft.Maui.Graphics.Point** to convert.

Return Value

The converted point

See Also

DiagramToView Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.DiagramToView Method (Rect)

Converts the specified rectangle from document to client coordinates.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Rect DiagramToView (
    Rect docRect
)
```

Visual Basic Copy Code

```
Public Function DiagramToView( _
    docRect As Rect _
) As Rect
```

Parameters

docRect
The **Microsoft.Maui.Graphics.Rect** to convert.

Return Value

The converted rectangle

See Also

DiagramToView Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.DiagramToView Method (Single)

Converts the specified value from document to client coordinates.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int DiagramToView (
    float value
)
```

Visual Basic Copy Code

```
Public Function DiagramToView( _
    value As Single _
) As Integer
```

Parameters

value

Return Value

See Also

DiagramToView Method Overload List
DiagramView Members

DiagramView.DrawAdjustmentHandlesScript Method

Sets the name of a JavaScript function that should be invoked when item's adjustment handles must be custom drawn.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView DrawAdjustmentHandlesScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function DrawAdjustmentHandlesScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.EnterInplaceEditScript Method

Sets the name of a JavaScript function that should be invoked when when entering inplace-edit mode.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView EnterInplaceEditScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function EnterInplaceEditScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView.FromJson Method

Overload List

Name	Description
DiagramView.FromJson (String)	Creates a DiagramView object from JSON.
DiagramView.FromJson (String, String)	Creates a Diagram object from JSON.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.FromJson Method (String)

Creates a DiagramView object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static DiagramView FromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    json As String _
) As DiagramView
```

Parameters

json
The JSON string

Return Value

See Also

FromJson Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.FromJson Method (String, String)

Creates a Diagram object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public static DiagramView FromJson (
    string id,
    string json
)
```

Visual Basic

```
Public Shared Function FromJson( _
    id As String, _
    json As String _
) As DiagramView
```

Parameters

id
The Id of the DiagramView

json
The JSON string

Return Value

See Also

FromJson Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.FromRequest Method

Overload List

Name	Description
DiagramView.FromRequest (String, HttpRequest)	Creates an instance of DiagramView.
DiagramView.FromRequest (String, HttpRequestBase)	Creates an instance of DiagramView.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.FromRequest Method (String, HttpRequest)

Creates an instance of DiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public static DiagramView FromRequest (
    string id,
    HttpRequest request
)
```

Visual Basic

```
Public Shared Function FromRequest( _
    id As String, _
    request As HttpRequest _
) As DiagramView
```

Parameters

id

The Id of the DiagramView

request

An HttpRequest object that contains the json data for the view.

Return Value

See Also

FromRequest Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.FromRequest Method (String, HttpRequestBase)

Creates an instance of DiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static DiagramView FromRequest (
    string id,
    HttpRequestBase request
)
```

Visual Basic Copy Code

```
Public Shared Function FromRequest( _
    id As String, _
    request As HttpRequestBase _
) As DiagramView
```

Parameters

id

The Id of the DiagramView

request

An **HttpRequest** object that contains the json data for the view.

Return Value

See Also

FromRequest Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.GetAttributes Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
protected string GetAttributes ()
```

Visual Basic Copy Code

```
Protected Function GetAttributes () As String
```

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.HeaderResizedScript Method

Sets the name of a JavaScript function that should be invoked when the user finished resizing a header.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView HeaderResizedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function HeaderResizedScript( _
    value As String _
) As DiagramView
```

Parameters

value
A string specifying the name of the client-side event handler.

Return Value

Returns this instance for method chaining.

Remarks

This event is raised when AllowResizeHeaders is enabled and the user resizes a grid column or row.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.HeaderStartResizingScript Method

Sets the name of a JavaScript function that should be invoked when the user starts resizing a header.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView HeaderStartResizingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function HeaderStartResizingScript( _
    value As String _
) As DiagramView
```

Parameters

value

A string specifying the name of the client-side event handler.

Return Value

Returns this instance for method chaining.

Remarks

This is a validation event raised when AllowResizeHeaders is enabled and the user starts resizing a grid column or row.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.HitTestAdjustmentHandlesScript Method

Sets the name of a JavaScript function that should be invoked when custom hit testing of item's adjustment handles must be performed.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView HitTestAdjustmentHandlesScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function HitTestAdjustmentHandlesScript( _
    value As String _
) As DiagramView
```

Parameters

value

The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.HyperlinkClickedScript Method

Sets the name of a JavaScript function that should be invoked when the user clicks a hyperlink inside a node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView HyperlinkClickedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function HyperlinkClickedScript( _
    value As String _
) As DiagramView
```

Parameters

value
Name of the hyperlinkClicked event handler function.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.Init Method

Renders the DiagramView's initializing scripts on a web page under the specified function name.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public virtual string Init (
    string functionName
)
```

Visual Basic Copy Code

```
Public Overridable Function Init( _
    functionName As String _
) As String
```

Parameters

functionName

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.InitializeLinkScript Method

Sets the name of a JavaScript function that should be invoked when the user starts drawing a new link, just after the DiagramLink instance is created.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView InitializeLinkScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function InitializeLinkScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.InitializeNodeScript Method

Sets the name of a JavaScript function that should be invoked when the user starts drawing a new node, just after the DiagramNode instance is created.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView InitializeNodeScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function InitializeNodeScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ItemAddedScript Method

Sets the name of a JavaScript function that should be invoked when an item is added to the diagram.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView ItemAddedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ItemAddedScript( _
    value As String _
) As DiagramView
```

Parameters

value
A string specifying the name of the client-side event handler.

Return Value

Returns this instance for method chaining.

Remarks

This event is raised both when items are drawn interactively or added from code - unlike NodeCreated and LinkCreated which are only raised for interactively-created items.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ItemRemovedScript Method

Sets the name of a JavaScript function that should be invoked when an item is removed from the diagram.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView ItemRemovedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ItemRemovedScript( _
    value As String _
) As DiagramView
```

Parameters

value

A string specifying the name of the client-side event handler.

Return Value

Returns this instance for method chaining.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LaneGridCellTextEditedScript Method

Sets the name of a JavaScript function that should be invoked hen the used edits the text of a grid cell.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LaneGridCellTextEditedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LaneGridCellTextEditedScript( _
    value As String _
) As DiagramView
```

Parameters

value
Name of the laneGridCellTextEdited event handler function.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LeaveInplaceEditScript Method

Sets the name of a JavaScript function that should be invoked when when leaving inplace-edit mode.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LeaveInplaceEditScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LeaveInplaceEditScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkClickedScript Method

Sets the name of a JavaScript function that should be invoked when the users clicks a link.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkClickedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkClickedScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkCreatedScript Method

Sets the name of a JavaScript function that should be invoked when the user draws a new link.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkCreatedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkCreatedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkCreatingScript Method

Sets the name of a JavaScript function that should be invoked while the user is drawing a new link.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkCreatingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkCreatingScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkDeletedScript Method

Sets the name of a JavaScript function that should be invoked when a link is deleted, either programmatically or by the user.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkDeletedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkDeletedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkDeletingScript Method

Sets the name of a JavaScript function that should be invoked when the user tries to delete a link.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkDeletingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkDeletingScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkDeselectedScript Method

Sets the name of a JavaScript function that should be invoked when a link is deselected, either programmatically or by the user.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkDeselectedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkDeselectedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkDoubleClickedScript Method

Sets the name of a JavaScript function that should be invoked when the users double-clicks a link.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkDoubleClickedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkDoubleClickedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkModifiedScript Method

Sets the name of a JavaScript function that should be invoked when the user moves any control point of a link.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkModifiedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkModifiedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkModifyingScript Method

Sets the name of a JavaScript function that should be invoked while the user is modifying a link.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkModifyingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkModifyingScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkPastedScript Method

Sets the name of a JavaScript function that should be invoked when a diagram link is pasted from clipboard.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkPastedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkPastedScript( _
    value As String _
) As DiagramView
```

Parameters

value
A string specifying JavaScript function name.

Return Value

This instance returned for method chaining.

Remarks

The handler function receives a LinkEventArgs object containing more information about the event.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkPointedScript Method

Sets the name of a JavaScript function that should be invoked when the mouse pointer hovers over a link.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkPointedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkPointedScript( _
    value As String _
) As DiagramView
```

Parameters

value
Name of the linkPointed event handler function.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkSelectedScript Method

Sets the name of a JavaScript function that should be invoked when a link is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkSelectedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkSelectedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkSelectingScript Method

Sets the name of a JavaScript function that should be invoked when the user tries to select a link.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkSelectingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkSelectingScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkTextEditedScript Method

Sets the name of a JavaScript function that should be invoked when the user edits the text of a link.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkTextEditedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkTextEditedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LinkTextEditingScript Method

Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a link.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView LinkTextEditingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkTextEditingScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LoadFromJson Method

Overload List

Name	Description
DiagramView.LoadFromJson (String)	Loads the specified JSON data.
DiagramView.LoadFromJson (String, Boolean)	Loads the specified JSON data.

See Also

DiagramView.LoadFromJson Method (String)

Loads the specified JSON data.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void LoadFromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Sub LoadFromJson( _
    json As String _
)
```

Parameters

json
The JSON string

See Also

LoadFromJson Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LoadFromJson Method (String, Boolean)

Loads the specified JSON data.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void LoadFromJson (
    string json,
    bool clearDiagram
)
```

Visual Basic Copy Code

```
Public Sub LoadFromJson( _
    json As String, _
    clearDiagram As Boolean _
)
```

Parameters

json
The JSON string

clearDiagram
true to remove all diagram items before loading, or **false** otherwise.

See Also

LoadFromJson Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LoadViewProps Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
protected virtual void LoadViewProps (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overridable Sub LoadViewProps( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

context

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MouseWheelScript Method

Sets the name of a JavaScript function that should be invoked when the user rotates the mouse wheel while the cursor is positioned over the diagram surface.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView MouseWheelScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function MouseWheelScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeClickedScript Method

Sets the name of a JavaScript function that should be invoked when the users clicks a node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeClickedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeClickedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeCreatedScript Method

Sets the name of a JavaScript function that should be invoked when the user draws a new node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeCreatedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeCreatedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeCreatingScript Method

Sets the name of a JavaScript function that should be invoked while the user is drawing a new node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeCreatingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeCreatingScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeDeletedScript Method

Sets the name of a JavaScript function that should be invoked when a node is deleted, either programmatically or by the user.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeDeletedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeDeletedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeDeletingScript Method

Sets the name of a JavaScript function that should be invoked when the user tries to delete a node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeDeletingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeDeletingScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeDeselectedScript Method

Sets the name of a JavaScript function that should be invoked when a node is deselected, either programmatically or by the user.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeDeselectedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeDeselectedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeDomCreatedScript Method

Sets the name of a JavaScript function that should be invoked when the content of a ControlNode is added to the page DOM.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeDomCreatedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeDomCreatedScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeDoubleClickedScript Method

Sets the name of a JavaScript function that should be invoked when the users double-clicks a node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeDoubleClickedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeDoubleClickedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeModifiedScript Method

Sets the name of a JavaScript function that should be invoked when the user moves or resizes a node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeModifiedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeModifiedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeModifyingScript Method

Sets the name of a JavaScript function that should be invoked while the user is moving or resizing a node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeModifyingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeModifyingScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodePastedScript Method

Sets the name of a JavaScript function that should be invoked when a diagram node is pasted from clipboard.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodePastedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodePastedScript( _
    value As String _
) As DiagramView
```

Parameters

value
A string specifying JavaScript function name.

Return Value

This instance returned for method chaining.

Remarks

The handler function receives a NodeEventArgs object containing more information about the event.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodePointedScript Method

Sets the name of a JavaScript function that should be invoked when the mouse pointer hovers over a node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodePointedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodePointedScript( _
    value As String _
) As DiagramView
```

Parameters

value
Name of the nodePointed event handler function.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeSelectedScript Method

Sets the name of a JavaScript function that should be invoked when a node is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeSelectedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeSelectedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeSelectingScript Method

Sets the name of a JavaScript function that should be invoked when the user tries to select a node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeSelectingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeSelectingScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeTextEditedScript Method

Sets the name of a JavaScript function that should be invoked when the user edits the text of a node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeTextEditedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeTextEditedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.NodeTextEditingScript Method

Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView NodeTextEditingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeTextEditingScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.RegisterConverters Method

Registers custom converters used for custom item types serialization.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView RegisterConverters (
    JavaScriptConverter[] converters
)
```

Visual Basic Copy Code

```
Public Function RegisterConverters( _
    converters() As JavaScriptConverter _
) As DiagramView
```

Parameters

converters
The converters to register.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.RegisterItemType Method

Overload List

Name	Description
DiagramView.RegisterItemType (Type, String, String, Int32)	Registers a custom item class for serialization support.
DiagramView.RegisterItemType (Type, String, String, Int32, String)	Registers a custom item class for serialization support.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.RegisterItemType Method (Type, String, String, Int32)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView RegisterItemType (
    Type itemClass,
    string classId,
    string clientClassId,
    int classVersion
)
```

Visual Basic Copy Code

```
Public Function RegisterItemType( _
    itemClass As Type, _
    classId As String, _
    clientClassId As String, _
    classVersion As Integer _
) As DiagramView
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when serializing, saving and loading items of the specified type.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

classVersion
A revision number of the item's class serialization format.

Return Value

See Also

RegisterItemType Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.RegisterItemType Method (Type, String, String, Int32, String)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView RegisterItemType (
    Type itemClass,
    string classId,
    string clientClassId,
    int classVersion,
    string jsonTemplate
)
```

Visual Basic — Copy Code

```
Public Function RegisterItemType( _
    itemClass As Type, _
    classId As String, _
    clientClassId As String, _
    classVersion As Integer, _
    jsonTemplate As String _
) As DiagramView
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when serializing, saving and loading items of the specified type.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

classVersion
A revision number of the item's class serialization format.

jsonTemplate
A JSON string representation of the node contents.

Return Value

See Also

RegisterItemType Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.RegisterSession Method

Overload List

Name	Description
DiagramView.RegisterSession (HttpSessionStateBase)	Registers the session instance with the ItemTypeResolver used in custom item serialization.
DiagramView.RegisterSession (ISession)	Registers the session instance with the ItemTypeResolver used in custom item serialization.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.RegisterSession Method (HttpSessionStateBase)

Registers the session instance with the ItemTypeResolver used in custom item serialization.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# — Copy Code

```
public DiagramView RegisterSession (
    HttpSessionStateBase session
)
```

Visual Basic Copy Code

```
Public Function RegisterSession( _
    session As HttpSessionStateBase _
) As DiagramView
```

Parameters

session
An HttpSessionStateBase instance.

Return Value

This DiagramView instance for method chaining.

See Also

RegisterSession Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.RegisterSession Method (ISession)

Registers the session instance with the ItemTypeResolver used in custom item serialization.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView RegisterSession (
    ISession session
)
```

Visual Basic Copy Code

```
Public Function RegisterSession( _
    session As ISession _
) As DiagramView
```

Parameters

session
An ISession instance.

Return Value

See Also

RegisterSession Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.Render Method

Renders the DiagramView on a web page.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Render ()
```

Visual Basic Copy Code

```
Public Function Render () As String
```

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.RepaintScript Method

Sets the name of a JavaScript function that should be invoked when the diagram has been repainted.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView RepaintScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function RepaintScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SaveToJson Method

Serializes the DiagramView's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public virtual string SaveToJson ()
```

Visual Basic

```
Public Overridable Function SaveToJson () As String
```

Return Value

The JSON string.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SaveViewProps Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
protected virtual void SaveViewProps (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic

```
Protected Overridable Sub SaveViewProps( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

context

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SelectionChangedScript Method

Sets the name of a JavaScript function that should be invoked when the diagram selection has changed.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public DiagramView SelectionChangedScript (
    string value
)
```

Visual Basic

```
Public Function SelectionChangedScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SelectionModifyingScript Method

Sets the name of a JavaScript function that should be invoked while the user is moving or resizing multiple items.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SelectionModifyingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function SelectionModifyingScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SerializeTagScript Method

Sets the name of a JavaScript function that should be invoked when the Tag or Id of a diagram item must be serialized in XML format.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SerializeTagScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function SerializeTagScript( _
    value As String _
) As DiagramView
```

Parameters

value
Name of the serializeTag event handler function.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetAddScriptReferences Method

Sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetAddScriptReferences (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAddScriptReferences( _
    value As Boolean _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetAllowInplaceEdit Method

Sets a value indicating whether in-place editing of the text of objects is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetAllowInplaceEdit (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAllowInplaceEdit( _
    value As Boolean _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetAutoScroll Method

Sets a value indicating whether auto scrolling of the document area is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetAutoScroll (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAutoScroll( _
    value As Boolean _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetBehavior Method

Sets a value indicating how the control responds to user actions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetBehavior (
    Behavior value
)
```

Visual Basic Copy Code

```
Public Function SetBehavior( _
    value As Behavior _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetCustomLinkType Method

Sets the type of a MindFusion.Diagramming.DiagramLink derived class whose instance should be created when a user starts drawing.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetCustomLinkType (
    Type value
)
```

Visual Basic Copy Code

```
Public Function SetCustomLinkType( _
    value As Type _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetCustomNodeType Method

Sets the type of a MindFusion.Diagramming.DiagramNode derived class whose instance should be created when a user starts drawing.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetCustomNodeType (
    Type value
)
```

Visual Basic Copy Code

```
Public Function SetCustomNodeType( _
    value As Type _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetDelKeyAction Method

Sets a value indicating the action that should be performed when the user hits the Del key.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetDelKeyAction (
    DelKeyAction value
)
```

Visual Basic Copy Code

```
Public Function SetDelKeyAction( _
    value As DelKeyAction _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetEnabled Method

Sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetEnabled (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetEnabled( _
    value As Boolean _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetEnableMultiTouch Method

Sets a value indicating whethr the control handles DOM pointer events to implement multi-touch interactions, such as zooming, node rotation or simultaneous drawing of multiple diagram items.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetEnableMultiTouch (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetEnableMultiTouch( _
    value As Boolean _
) As DiagramView
```

Parameters

value
true to enable multi-touch, or **false** otherwise.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetFallbackContent Method

Sets the HTML to be rendered as the canvas' fallback content.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetFallbackContent (
    string value
)
```

Visual Basic Copy Code

```
Public Function SetFallbackContent( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetHeight Method

Sets the height of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetHeight (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetHeight( _
    value As Integer _
) As DiagramView
```

Parameters

value
The height of the view in pixels.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMagnifierEnabled Method

Sets a value indicating whether the magnifier tool is currently enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMagnifierEnabled (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetMagnifierEnabled( _
    value As Boolean _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMagnifierFactor Method

Sets the zoom factor of the magnifier tool.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMagnifierFactor (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetMagnifierFactor( _
    value As Single _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMagnifierFrameColor Method

Sets the color of the magnifier frame.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMagnifierFrameColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetMagnifierFrameColor( _
    value As Color _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMagnifierFrameThickness Method

Sets the thickness of the magnifier frame, expressed in pixels.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMagnifierFrameThickness (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetMagnifierFrameThickness( _
    value As Integer _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMagnifierHeight Method

Sets the height of the magnifier tool, expressed in pixels.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMagnifierHeight (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetMagnifierHeight( _
    value As Integer _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMagnifierSecondaryFrameColor Method

Sets the secondary color of the magnifier frame.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMagnifierSecondaryFrameColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetMagnifierSecondaryFrameColor( _
    value As Color _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMagnifierShading Method

Sets a value indicating whether to enhance the effect of a magnifier lense by using a gradient shading.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMagnifierShading (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetMagnifierShading( _
    value As Boolean _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMagnifierShape Method

Sets the shape of the magnifier tool.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMagnifierShape (
    Shape value
)
```

Visual Basic Copy Code

```
Public Function SetMagnifierShape( _
    value As Shape _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMagnifierWidth Method

Sets the width of the magnifier tool, expressed in pixels.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMagnifierWidth (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetMagnifierWidth( _
    value As Integer _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMaxJsonLength Method

Sets the maximum length of JSON strings that are accepted by internal JavaScriptSerializer instances.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMaxJsonLength (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetMaxJsonLength( _
    value As Integer _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMinVisibleFontSize Method

Sets a threshold value that hides text if scaled font sizes become smaller.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMinVisibleFontSize (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetMinVisibleFontSize( _
    value As Single _
) As DiagramView
```

Parameters

value
A **float** value specifying the smallest visible font size.

Return Value

This DiagramView instance returned for chaining calls.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetModificationStart Method

Sets how users can start modifying diagram items.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetModificationStart (
    ModificationStart value
)
```

Visual Basic Copy Code

```
Public Function SetModificationStart( _
    value As ModificationStart _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMouseWheelAction Method

Sets a value indicating what action to perform when users rotate the mouse wheel.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMouseWheelAction (
    MouseWheelAction value
)
```

Visual Basic Copy Code

```
Public Function SetMouseWheelAction( _
    value As MouseWheelAction _
) As DiagramView
```

Parameters

value
A member of the MouseWheelAction enumeration.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace
MouseWheelAction Property

DiagramView.SetMultiTouchDraw Method

Sets a value indicating whether the control allows drawing multiple items simultaneously.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMultiTouchDraw (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetMultiTouchDraw( _
    value As Boolean _
) As DiagramView
```

Parameters

value
true to allow drawing multiple items simultaneously, or **false** otherwise.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetMultiTouchModify Method

Sets a value indicating whether multi-touch modifications are enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMultiTouchModify (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetMultiTouchModify( _
    value As Boolean _
) As DiagramView
```

Parameters

value
true to enable multi-touch node modification, or **false** otherwise.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace
MultiTouchModify Property

DiagramView.SetMultiTouchZoom Method

Sets a value indicating whether multi-touch zoom is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetMultiTouchZoom (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetMultiTouchZoom( _
    value As Boolean _
) As DiagramView
```

Parameters

value
true to enable multi-touch zoom, or **false** otherwise.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace
MultiTouchZoom Property

DiagramView.SetRecursionLimit Method

Sets the limit for constraining the number of object levels to process by internal JavaScriptSerializer instances.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetRecursionLimit (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetRecursionLimit( _
    value As Integer _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetScrollLeft Method

Sets the view's X scroll position.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetScrollLeft (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetScrollLeft( _
    value As Single _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetScrollTop Method

Sets the view's Y scroll position.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetScrollTop (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetScrollTop( _
    value As Single _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetShowScrolbars Method

Sets a value indicating whether the scrollbars are visible.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetShowScrolbars (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetShowScrolbars( _
    value As Boolean _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetTooltipDelay Method

Sets the delay before a tooltip is shown.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetTooltipDelay (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetTooltipDelay( _
    value As Integer _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetVirtualScroll Method

Sets a value indicating whether virtual scroll mode is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetVirtualScroll (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetVirtualScroll( _
    value As Boolean _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetWidth Method

Sets the width of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetWidth (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetWidth( _
    value As Integer _
) As DiagramView
```

Parameters

value
The width of the view in pixels.

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SetZoomFactor Method

Sets the zoom of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SetZoomFactor (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetZoomFactor( _
    value As Single _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.SizeChangedScript Method

Sets the name of a JavaScript function that should be invoked when the size of the diagram canvas has changed.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView SizeChangedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function SizeChangedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.TableColumnResizedScript Method

Sets the name of a JavaScript function that should be invoked when the user finishes resizing a column in a table interactively.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView TableColumnResizedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function TableColumnResizedScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.TableColumnResizingScript Method

Sets the name of a JavaScript function that should be invoked when the user resizes a column in a table interactively.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView TableColumnResizingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function TableColumnResizingScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.TableRowResizedScript Method

Sets the name of a JavaScript function that should be invoked when the user finishes resizing a row in a table interactively.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView TableRowResizedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function TableRowResizedScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.TableRowResizingScript Method

Sets the name of a JavaScript function that should be invoked when the user resizes a row in a table interactively.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView TableRowResizingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function TableRowResizingScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ToJson Method

Serializes the DiagramView's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static string ToJson (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public Shared Function ToJson( _
    view As DiagramView _
) As String
```

Parameters

view
An instance of the DiagramView class.

Return Value

The JSON string.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.TreeCollapsedScript Method

Sets the name of a JavaScript function that should be invoked when a user collapses a tree branch by clicking the [+] button of an Expandable node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView TreeCollapsedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function TreeCollapsedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.TreeExpandedScript Method

Sets the name of a JavaScript function that should be invoked when a user expands a tree branch by clicking the [+] button of an Expandable node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView TreeExpandedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function TreeExpandedScript( _
    value As String _
) As DiagramView
```

Parameters

value
The name of the function

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.TreeItemTextEditedScript Method

Sets the name of a JavaScript function that should be invoked when the user edits the text of a tree view item.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView TreeItemTextEditedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function TreeItemTextEditedScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.TreeItemTextEditingScript Method

Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a tree view item.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView TreeItemTextEditingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function TreeItemTextEditingScript( _
    value As String _
) As DiagramView
```

Parameters

value

Return Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ViewToDiagram Method

Overload List

Name	Description
DiagramView.ViewToDiagram (Int32)	Converts the specified value from client to document coordinates.
DiagramView.ViewToDiagram (Point)	Converts the specified point from client to document coordinates.
DiagramView.ViewToDiagram (Rect)	Converts the specified rectangle from client to document coordinates.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ViewToDiagram Method (Int32)

Converts the specified value from client to document coordinates.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float ViewToDiagram (
    int value
)
```

Visual Basic Copy Code

```
Public Function ViewToDiagram( _
    value As Integer _
) As Single
```

Parameters

value

Return Value

See Also

ViewToDiagram Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ViewToDiagram Method (Point)

Converts the specified point from client to document coordinates.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Point ViewToDiagram (
    Point clientPoint
)
```

Visual Basic Copy Code

```
Public Function ViewToDiagram( _
    clientPoint As Point _
) As Point
```

Parameters

clientPoint

Return Value

See Also

ViewToDiagram Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ViewToDiagram Method (Rect)

Converts the specified rectangle from client to document coordinates.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Rect ViewToDiagram (
    Rect clientRect
)
```

Visual Basic Copy Code

```
Public Function ViewToDiagram( _
    clientRect As Rect _
) As Rect
```

Parameters

clientRect

Return Value

See Also

ViewToDiagram Method Overload List
DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView Properties

DiagramView.AddScriptReferences Property

Gets or sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AddScriptReferences { get; set; }
```

Visual Basic Copy Code

```
Public Property AddScriptReferences As Boolean
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.AllowInplaceEdit Property

Gets or sets a value indicating whether in-place editing of the text of objects is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AllowInplaceEdit { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowInplaceEdit As Boolean
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.AutoClasses Property

Gets the registered client types with JSON templates.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Dictionary<string, string> AutoClasses { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property AutoClasses As Dictionary(Of String, String)
```

Property Value

A **Dictionary** containing registered JSON templates for automatic client-side node types.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.AutoScroll Property

Gets or sets a value indicating whether auto scrolling of the document area is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AutoScroll { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoScroll As Boolean
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.Behavior Property

Gets or sets a value indicating how the control responds to user actions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Behavior Behavior { get; set; }
```

Visual Basic Copy Code

```
Public Property Behavior As Behavior
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ClientScriptLocation Property

Gets or sets the URL to the MindFusion.Diagramming.js file.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string ClientScriptLocation { get; set; }
```

Visual Basic Copy Code

```
Public Property ClientScriptLocation As String
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.Converters Property

Gets the registered custom converters.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public JavaScriptConverter[] Converters { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Converters As JavaScriptConverter()
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.CustomLinkType Property

Gets the type of a MindFusion.Diagramming.DiagramLink derived class whose instance should be created when a user starts drawing.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Type CustomLinkType { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property CustomLinkType As Type
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.CustomNodeType Property

Gets the type of a MindFusion.Diagramming.DiagramNode derived class whose instance should be created when a user starts drawing.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Type CustomNodeType { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property CustomNodeType As Type
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.DefaultControlTemplate Property

Gets or sets the HTML string representing the DOM content, rendered by default in newly created ControlNode objects.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string DefaultControlTemplate { get; set; }
```

Visual Basic Copy Code

```
Public Property DefaultControlTemplate As String
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.DelKeyAction Property

Gets or sets a value indicating the action that should be performed when the user hits the DELETE key.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DelKeyAction DelKeyAction { get; set; }
```

Visual Basic Copy Code

```
Public Property DelKeyAction As DelKeyAction
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.Diagram Property

Gets or sets the Diagram object displayed by this view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Diagram Diagram { get; set; }
```

Visual Basic Copy Code

```
Public Property Diagram As Diagram
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.Enabled Property

Gets or sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool Enabled { get; set; }
```

Visual Basic Copy Code

```
Public Property Enabled As Boolean
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.EnableMultiTouch Property

Gets or sets a value indicating whether the control handles DOM pointer events to implement multi-touch interactions, such as zooming, node rotation or simultaneous drawing of multiple diagram items.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool EnableMultiTouch { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableMultiTouch As Boolean
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.FallbackContent Property

Gets or sets the HTML to be rendered as the canvas' fallback content.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string FallbackContent { get; set; }
```

Visual Basic Copy Code

```
Public Property FallbackContent As String
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.HandleTouchHitDistance Property

Gets or sets maximum distance between touch position and adjustment handle at which the handle can still be grabbed to modify its item.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float HandleTouchHitDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property HandleTouchHitDistance As Single
```

Property Value

A **float** value specifying the allowed hit distance, in diagram's current unit. The default value is 5.

Remarks

This value is considered only when handling touch events.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.Height Property

Gets or sets the height of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Height { get; set; }
```

Visual Basic Copy Code

```
Public Property Height As Integer
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.Id Property

Gets or sets the id of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ItemTypes Property

Gets the registered item types.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Dictionary<Type, string> ItemTypes { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ItemTypes As Dictionary(Of Type, String)
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.LicenseKey Property

Gets or sets the license key of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string LicenseKey { get; set; }
```

Visual Basic Copy Code

```
Public Property LicenseKey As String
```

Property Value

A string containing the license key.

Remarks

Set this property to disable the component's evaluation mode and stop displaying trial messages. If your application contains more than one Diagram instance or other controls by MindFusion, you could call the MindFusion.Licensing.LicenseManager.AddLicense method to specify the key once instead of setting it per each control.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MagnifierEnabled Property

Gets or sets a value indicating whether the magnifier tool is currently enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool MagnifierEnabled { get; set; }
```

Visual Basic Copy Code

```
Public Property MagnifierEnabled As Boolean
```

Property Value

See Also

DiagramView Members
DiagramView Class

DiagramView.MagnifierFactor Property

Gets or sets the zoom factor of the magnifier tool.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float MagnifierFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property MagnifierFactor As Single
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MagnifierFrameColor Property

Gets or sets the color of the magnifier frame.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color MagnifierFrameColor { get; set; }
```

Visual Basic Copy Code

```
Public Property MagnifierFrameColor As Color
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MagnifierFrameThickness Property

Gets or sets the thickness of the magnifier frame, expressed in pixels.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int MagnifierFrameThickness { get; set; }
```

Visual Basic Copy Code

```
Public Property MagnifierFrameThickness As Integer
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MagnifierHeight Property

Gets or sets the height of the magnifier tool, expressed in pixels.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int MagnifierHeight { get; set; }
```

Visual Basic Copy Code

```
Public Property MagnifierHeight As Integer
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MagnifierSecondaryFrameColor Property

Gets or sets the secondary color of the magnifier frame.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color MagnifierSecondaryFrameColor { get; set; }
```

Visual Basic Copy Code

```
Public Property MagnifierSecondaryFrameColor As Color
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MagnifierShading Property

Gets or sets a value indicating whether to enhance the effect of a magnifier lense by using a gradient shading.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool MagnifierShading { get; set; }
```

Visual Basic Copy Code

```
Public Property MagnifierShading As Boolean
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MagnifierShape Property

Gets or sets the shape of the magnifier tool.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Shape MagnifierShape { get; set; }
```

Visual Basic Copy Code

```
Public Property MagnifierShape As Shape
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MagnifierWidth Property

Gets or sets the width of the magnifier tool, expressed in pixels.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int MagnifierWidth { get; set; }
```

Visual Basic Copy Code

```
Public Property MagnifierWidth As Integer
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MaxJsonLength Property

Gets or sets the maximum length of JSON strings that are accepted by internal JavaScriptSerializer instances.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static int MaxJsonLength { get; set; }
```

Visual Basic Copy Code

```
Public Shared Property MaxJsonLength As Integer
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MinVisibleFontSize Property

Gets or sets a threshold value that hides text if scaled font sizes become smaller.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float MinVisibleFontSize { get; set; }
```

Visual Basic Copy Code

```
Public Property MinVisibleFontSize As Single
```

Property Value

A **float** value specifying the smallest visible font size.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ModificationStart Property

Gets or sets how users can start modifying diagram items.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ModificationStart ModificationStart { get; set; }
```

Visual Basic Copy Code

```
Public Property ModificationStart As ModificationStart
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MouseWheelAction Property

Gets or sets what action to perform when users rotate the mouse wheel.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public MouseWheelAction MouseWheelAction { get; set; }
```

Visual Basic Copy Code

```
Public Property MouseWheelAction As MouseWheelAction
```

Property Value

A member of the MouseWheelAction enumeration. The default value is **Scroll**.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace
MouseWheelScript Method

DiagramView.MultiTouchDraw Property

Gets or sets a value indicating whether the control allows drawing multiple items simultaneously.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool MultiTouchDraw { get; set; }
```

Visual Basic

```
Public Property MultiTouchDraw As Boolean
```

Property Value

true to allow drawing multiple items simultaneously, or **false** otherwise. The default value is **true**.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.MultiTouchModify Property

Gets or sets a value indicating whether multi-touch modifications are enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public bool MultiTouchModify { get; set; }
```

Visual Basic

```
Public Property MultiTouchModify As Boolean
```

Property Value

true to enable multi-touch node modification, or **false** otherwise. The default value is **true**.

Remarks

If this property is enabled, two touches on same node will scale and rotate it. Otherwise each touch is handled independently according to current Behavior value, e.g. by drawing a new diagram link.

Multi-touch rotation is applied only if the Rotate handle bit in EnabledHandles is set.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace
MultiTouchZoom Property

DiagramView.MultiTouchZoom Property

Gets or sets a value indicating whether multi-touch zoom is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public bool MultiTouchZoom { get; set; }
```

Visual Basic

```
Public Property MultiTouchZoom As Boolean
```

Property Value

true to enable multi-touch zoom, or **false** otherwise. The default value is **true**.

Remarks

If this property is disabled, each touch is handled independently according to current Behavior value, e.g. by drawing a new diagram item or moving an existing one.

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace
MultiTouchModify Property

DiagramView.RecursionLimit Property

Gets or sets the limit for constraining the number of object levels to process by internal JavaScriptSerializer instances.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static int RecursionLimit { get; set; }
```

Visual Basic Copy Code

```
Public Shared Property RecursionLimit As Integer
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ScrollLeft Property

Gets or sets the view's X scroll position.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float ScrollLeft { get; set; }
```

Visual Basic Copy Code

```
Public Property ScrollLeft As Single
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ScrollTop Property

Gets or sets the view's Y scroll position.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float ScrollTop { get; set; }
```

Visual Basic Copy Code

```
Public Property ScrollTop As Single
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ShapeLibraryLocation Property

The location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string ShapeLibraryLocation { get; set; }
```

Visual Basic Copy Code

```
Public Property ShapeLibraryLocation As String
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.TooltipDelay Property

Gets or sets the delay before a tooltip is shown.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int TooltipDelay { get; set; }
```

Visual Basic Copy Code

```
Public Property TooltipDelay As Integer
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.VirtualScroll Property

Gets or sets a value indicating whether virtual scroll mode is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool VirtualScroll { get; set; }
```

Visual Basic Copy Code

```
Public Property VirtualScroll As Boolean
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.Width Property

Gets or sets the width of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Width { get; set; }
```

Visual Basic Copy Code

```
Public Property Width As Integer
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramView.ZoomFactor Property

Gets or sets the zoom of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float ZoomFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property ZoomFactor As Single
```

Property Value

See Also

DiagramView Members
DiagramView Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewConverter Class

Serializes DiagramView instances.

Syntax

C# Copy Code

```
public class DiagramViewConverter : JavaScriptConverter
```

Visual Basic Copy Code

```
Public Class DiagramViewConverter
    Inherits JavaScriptConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.DiagramViewConverter

See Also

DiagramViewConverter Members
MindFusion.Diagramming.Mvc Namespace

DiagramViewConverter Members

Serializes DiagramView instances.

The following tables list the members exposed by the DiagramViewConverter type.

Public Constructors

	Name	Description
	DiagramViewConverter	

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

DiagramViewConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewConverter Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramViewConverter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramViewConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewConverter Methods

DiagramViewConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

DiagramViewConverter Members
DiagramViewConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

DiagramViewConverter Members
DiagramViewConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewConverter Properties

DiagramViewConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

DiagramViewConverter Members
DiagramViewConverter Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions Class

Contains extension methods of DiagramView.

Syntax

C# Copy Code

```
public sealed class DiagramViewExtensions
```

Visual Basic Copy Code

```
Public NotInheritable Class DiagramViewExtensions
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.DiagramViewExtensions

See Also

DiagramViewExtensions Members
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions Members

Contains extension methods of DiagramView.

The following tables list the members exposed by the DiagramViewExtensions type.

Public Methods

	Name	Description
	DiagramView	Overloaded. Returns a **DiagramView** by using the specified HTML helper.

See Also

DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions Methods

DiagramViewExtensions.DiagramView Method

Returns a DiagramView by using the specified HTML helper.

Overload List

Name	Description
DiagramViewExtensions.DiagramView (IHtmlHelper, DiagramView)	Returns a **DiagramView** by using the specified HTML helper.
DiagramViewExtensions.DiagramView (IHtmlHelper, DiagramView, IDictionary(String, Object))	Returns a **DiagramView** by using the specified HTML helper.
DiagramViewExtensions.DiagramView (IHtmlHelper, DiagramView, Object)	Returns a **DiagramView** by using the specified HTML helper.
DiagramViewExtensions.DiagramView (IHtmlHelper, DiagramView, String)	Returns a DiagramView by using the specified HTML helper.
DiagramViewExtensions.DiagramView (IHtmlHelper, DiagramView, String, IDictionary(String, Object))	Returns a **DiagramView** by using the specified HTML helper.
DiagramViewExtensions.DiagramView (IHtmlHelper, DiagramView, String, Object)	Returns a **DiagramView** by using the specified HTML helper.
DiagramViewExtensions.DiagramView (IHtmlHelper, String, Diagram)	Returns a **DiagramView** by using the specified HTML helper.
DiagramViewExtensions.DiagramView (IHtmlHelper, String, Diagram, IDictionary(String, Object))	Returns a **DiagramView** by using the specified HTML helper.
DiagramViewExtensions.DiagramView (IHtmlHelper, String, Diagram, Object)	Returns a **DiagramView** by using the specified HTML helper.
DiagramViewExtensions.DiagramView (IHtmlHelper, String, Diagram, String)	Returns a **DiagramView** by using the specified HTML helper.
DiagramViewExtensions.DiagramView (IHtmlHelper, String, Diagram, String, IDictionary(String, Object))	Returns a **DiagramView** by using the specified HTML helper.
DiagramViewExtensions.DiagramView (IHtmlHelper, String, Diagram, String, Object)	Returns a **DiagramView** by using the specified HTML helper.

See Also

DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions.DiagramView Method (IHtmlHelper, DiagramView)

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    DiagramView diagramView
)
```

Visual Basic Copy Code

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    diagramView As DiagramView _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

diagramView
The DiagramView.

Return Value

See Also

DiagramView Method Overload List
DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions.DiagramView Method (IHtmlHelper, DiagramView, IDictionary(String, Object))

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    DiagramView diagramView,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    diagramView As DiagramView, _
    htmlAttributes As IDictionary(Of String, Object) _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

diagramView
The DiagramView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

Remarks

Use the **htmlAttributes** parameter to pass style data to the control's outer div element. The following example sets the control's width and height properties and hides the scrollbars.

Example

HTML

```
@Html.DiagramView(
    ((DiagramView)ViewBag.DiagramView),
    new { style = "width:500px;height:500px;overflow:hidden;"})
```

See Also

DiagramView Method Overload List
DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions.DiagramView Method (IHtmlHelper, DiagramView, Object)

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    DiagramView diagramView,
    Object htmlAttributes
)
```

Visual Basic

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    diagramView As DiagramView, _
    htmlAttributes As Object _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

diagramView
The DiagramView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

Remarks

Use the **htmlAttributes** parameter to pass style data to the control's outer div element. The following example sets the control's width and height properties and hides the scrollbars.

Example

HTML

```
@Html.DiagramView(
    ((DiagramView)ViewBag.DiagramView),
    new { style = "width:500px;height:500px;overflow:hidden;"})
```

See Also

DiagramView Method Overload List
DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions.DiagramView Method (IHtmlHelper, DiagramView, String)

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    DiagramView diagramView,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    diagramView As DiagramView, _
    initFunction As String _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

diagramView
The DiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

DiagramView Method Overload List
DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions.DiagramView Method (IHtmlHelper, DiagramView, String, IDictionary(String, Object))

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    DiagramView diagramView,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    diagramView As DiagramView, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

diagramView
The DiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

DiagramView Method Overload List
DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions.DiagramView Method (IHtmlHelper, DiagramView, String, Object)

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    DiagramView diagramView,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    diagramView As DiagramView, _
    initFunction As String, _
    htmlAttributes As Object _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

diagramView
The DiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

DiagramView Method Overload List
DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions.DiagramView Method (IHtmlHelper, String, Diagram)

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    string id,
    Diagram diagram
)
```

Visual Basic Copy Code

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    id As String, _
    diagram As Diagram _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

id
The Id of the DiagramView.

diagram
The Diagram displayed by the view.

Return Value

See Also

DiagramView Method Overload List
DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions.DiagramView Method (IHtmlHelper, String, Diagram, IDictionary(String, Object))

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    string id,
    Diagram diagram,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    id As String, _
    diagram As Diagram, _
    htmlAttributes As IDictionary(Of String, Object) _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

id
The Id of the DiagramView

diagram
The Diagram displayed by the view

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

DiagramView Method Overload List
DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions.DiagramView Method (IHtmlHelper, String, Diagram, Object)

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    string id,
    Diagram diagram,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    id As String, _
    diagram As Diagram, _
    htmlAttributes As Object _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

id
The Id of the DiagramView

diagram
The Diagram displayed by the view

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

DiagramView Method Overload List
DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions.DiagramView Method (IHtmlHelper, String, Diagram, String)

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    string id,
    Diagram diagram,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    id As String, _
    diagram As Diagram, _
    initFunction As String _
) As MvcHtmlString
```

Parameters

helper

The HTML helper

id
The Id of the DiagramView

diagram
The Diagram displayed by the view

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

DiagramView Method Overload List
DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewExtensions.DiagramView Method (IHtmlHelper, String, Diagram, String, IDictionary(String, Object))

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    string id,
    Diagram diagram,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    id As String, _
    diagram As Diagram, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

id
The Id of the DiagramView

diagram
The Diagram displayed by the view

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

DiagramView Method Overload List
DiagramViewExtensions Members

DiagramViewExtensions.DiagramView Method (IHtmlHelper, String, Diagram, String, Object)

Returns a DiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString DiagramView (
    HtmlHelper helper,
    string id,
    Diagram diagram,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function DiagramView( _
    helper As HtmlHelper, _
    id As String, _
    diagram As Diagram, _
    initFunction As String, _
    htmlAttributes As Object _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

id
The Id of the DiagramView

diagram
The Diagram displayed by the view

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

DiagramView Method Overload List
DiagramViewExtensions Members
DiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewTagHelper Class

Tag helper of DiagramView.

Syntax

C# Copy Code

```
public class DiagramViewTagHelper : TagHelper
```

Visual Basic	Copy Code

```
Public Class DiagramViewTagHelper
    Inherits TagHelper
```

Remarks

This class implements the <mindfusion-diagramview> tag.

Inheritance Hierarchy

System.Object
  Microsoft.AspNetCore.Razor.TagHelpers.TagHelper
    MindFusion.Diagramming.Mvc.DiagramViewTagHelper

See Also

DiagramViewTagHelper Members
MindFusion.Diagramming.Mvc Namespace

DiagramViewTagHelper Members

Tag helper of DiagramView.

The following tables list the members exposed by the DiagramViewTagHelper type.

Public Constructors

	Name	Description
	DiagramViewTagHelper	

Public Properties

	Name	Description
	Component	
	Id	
	Style	

Public Methods

	Name	Description
	Process	

See Also

DiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewTagHelper Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#	Copy Code

```
public DiagramViewTagHelper ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

DiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewTagHelper Methods

DiagramViewTagHelper.Process Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override void Process (
    TagHelperContext context,
    TagHelperOutput output
)
```

Visual Basic Copy Code

```
Public Overrides Sub Process( _
    context As TagHelperContext, _
    output As TagHelperOutput _
)
```

Parameters

context

output

See Also

DiagramViewTagHelper Members
DiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewTagHelper Properties

DiagramViewTagHelper.Component Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramView Component { get; set; }
```

Visual Basic Copy Code

```
Public Property Component As DiagramView
```

Property Value

See Also

DiagramViewTagHelper Members
DiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewTagHelper.Id Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

DiagramViewTagHelper Members
DiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

DiagramViewTagHelper.Style Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As String
```

Property Value

See Also

DiagramViewTagHelper Members
DiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

FreeFormNodeConverter Class

Serializes FreeFormNode instances.

Syntax

C# Copy Code

```
public class FreeFormNodeConverter : DiagramNodeConverter
```

Visual Basic Copy Code

```
Public Class FreeFormNodeConverter
    Inherits DiagramNodeConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.DiagramItemConverter
      MindFusion.Diagramming.Mvc.DiagramNodeConverter
        MindFusion.Diagramming.Mvc.FreeFormNodeConverter

See Also

FreeFormNodeConverter Members
MindFusion.Diagramming.Mvc Namespace

FreeFormNodeConverter Members

Serializes FreeFormNode instances.

The following tables list the members exposed by the FreeFormNodeConverter type.

Public Constructors

	Name	Description
	FreeFormNodeConverter	Creates a new instance of the FreeFormNodeConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

FreeFormNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

FreeFormNodeConverter Constructor

Creates a new instance of the FreeFormNodeConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public FreeFormNodeConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

FreeFormNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

FreeFormNodeConverter Methods

FreeFormNodeConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

FreeFormNodeConverter Members
FreeFormNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

FreeFormNodeConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

FreeFormNodeConverter Members
FreeFormNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

FreeFormNodeConverter Properties

FreeFormNodeConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

FreeFormNodeConverter Members
FreeFormNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ItemListView Class

ListBox control with support for hosting DiagramItems and dragging them to the Diagram control.

Syntax

C# Copy Code

```
public class ItemListView
```

Visual Basic Copy Code

```
Public Class ItemListView
```

Remarks

Call the AddItem method to add an item to the view.

Drag-and-drop operations create a copy of the dragged item, and set its size to NewInstanceSize. The copy is created by calling item's clone method. You can override **clone** to copy property values of custom item subclasses.

Icons in the list are drawn by scaling **NewInstanceSize** down to IconSize. E.g. if a list icon is half the size of the diagram item it creates, its graphic elements such as cells, labels, strokes would be rendered at half scale too (as opposed to rendering with same label / stroke sizes but in smaller bounding rectangle). The scaling code assumes MeasureUnit of the target diagram and the list view have matching values.

Configuring the control

ItemListView can be created inside a Razor page either by invoking the ItemListView html helper, or by adding a <mindfusion-itemlistview> tag helper:

Razor Copy Code

```
@addTagHelper *, MindFusion.Diagramming.Mvc
....
<mindfusion-itemlistview
    component=(ItemListView)ViewBag.ItemListView
    style="width:100%; height:100%;">
</mindfusion-itemlistview>
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.ItemListView

See Also

ItemListView Members
MindFusion.Diagramming.Mvc Namespace
Palette Class

ItemListView Members

ListBox control with support for hosting DiagramItems and dragging them to the Diagram control.

The following tables list the members exposed by the ItemListView type.

Public Constructors

	Name	Description
	ItemListView	Overloaded.

Public Properties

	Name	Description
	AddScriptReferences	Gets or sets a value indicating whether the control should render references to the client scripts.
	AutoClasses	Gets the registered client types with JSON templates.
	AutoConnectDistance	Gets or sets the maximum distance to a node when auto connecting links.

	AutoConnectLinks	Gets or sets whether links should be automatically connected to the nearest node.
	BackColor	Gets or sets the background color of the ItemListView control.
	ClientScriptLocation	Gets or sets the URL to the MindFusion.Diagramming.js file.
	Columns	Gets or sets the number of columns in control's layout.
	Enabled	Gets or sets a value indicating whether handling of mouse events is enabled.
	Font	Gets or sets the font of the items captions.
	IconSize	Gets or sets the size of the items displayed in the view.
	Id	Gets or sets the id of the view.
	Items	Gets or sets the collection of items displayed in the view.
	LabelPosition	Gets or sets a value indicating the position of labels relative to item icons.
	MeasureUnit	Gets or sets the unit of measure used to specify item sizes.
	MinVisibleFontSize	Gets or sets a threshold value that hides text if scaled font sizes become smaller.
	NewInstanceSize	Gets or sets the default size of items created by drag-and-drop from this list view.
	Orientation	Gets or sets the layout orientation of the control.
	Padding	Gets or sets the padding space between item's slot bounds and its contents.
	ShapeLibraryLocation	The location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path.
	TextColor	Gets or sets the color of the items captions.

Public Methods

	Name	Description
	AddItem	Overloaded.
	ControlLoadedScript	Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.
S	FromJson	Overloaded.
S	FromRequest	Initializes a new instance of the ItemListView class.
	Init	Renders the ItemListView's initializing scripts on a web page under the specified function name.
	LinkSelectedScript	Sets the name of a JavaScript function that should be invoked when a link is selected, either programmatically or by the user.
	LoadFromJson	Loads the specified JSON data.
	NodeSelectedScript	Sets the name of a JavaScript function that should be invoked when a node is selected, either programmatically or by the user.
	RegisterItemType	Overloaded.
	RegisterSession	Registers the session instance with the ItemTypeResolver used in custom item serialization.
	RemoveItem	Removes a DiagramItem from the displayed items collection.
	Render	Renders the ItemListView on a web page.

	SaveToJson	Serializes the ItemListView's data to JSON.
	SetAddScriptReferences	Sets a value indicating whether the control should render references to the client scripts.
	SetAutoConnectDistance	Sets the maximum distance to a node when auto connecting links.
	SetAutoConnectLinks	Sets whether links should be automatically connected to the nearest node.
	SetBackColor	Sets the background color of the ItemListView control.
	SetColumns	Sets the number of columns in control's layout.
	SetEnabled	Sets a value indicating whether handling of mouse events is enabled.
	SetFont	Sets the font of the items captions.
	SetIconSize	Sets the size of the items displayed in the view.
	SetLabelPosition	Sets a value indicating the position of labels relative to item icons.
	SetMinVisibleFontSize	Sets a threshold value that hides text if scaled font sizes become smaller.
	SetNewInstanceSize	Sets the default size with which items dragged onto a diagram will be created.
	SetOrientation	Sets the layout orientation of the control.
	SetTextColor	Sets the text color of the items captions.
	ShapeLibraryLoadedScript	Sets the name of a JavaScript function that should be invoked when a shape library is loaded.
	ToJson	Serializes the ItemListView's data to JSON.

See Also

ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView Constructor

Overload List

Name	Description
ItemListView ()	Initializes a new instance of the ItemListView class.
ItemListView (String)	Initializes a new instance of the ItemListView class.
ItemListView (String, DiagramItemCollection)	Initializes a new instance of the ItemListView class.

See Also

ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView Constructor ()

Initializes a new instance of the ItemListView class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ItemListView Constructor Overload List
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView Constructor (String)

Initializes a new instance of the ItemListView class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView (
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String _
)
```

Parameters

id
The Id of the ItemListView

See Also

ItemListView Constructor Overload List
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView Constructor (String, DiagramItemCollection)

Initializes a new instance of the ItemListView class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView (
    string id,
    DiagramItemCollection items
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String, _
    items As DiagramItemCollection _
)
```

Parameters

id

The Id of the ItemListView

items

The collection of items to be displayed in the ItemListView

See Also

ItemListView Constructor Overload List
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView Methods

ItemListView.AddItem Method

Overload List

Name	Description
ItemListView.AddItem (DiagramItem)	Adds a DiagramItem to the displayed items collection.
ItemListView.AddItem (DiagramItem, String)	Adds a DiagramItem to the displayed items collection.

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.AddItem Method (DiagramItem)

Adds a DiagramItem to the displayed items collection.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView AddItem (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Function AddItem( _
    item As DiagramItem _
) As ItemListView
```

Parameters

item

The DiagramItem instance to add.

Return Value

See Also

AddItem Method Overload List
ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.AddItem Method (DiagramItem, String)

Adds a DiagramItem to the displayed items collection.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView AddItem (
    DiagramItem item,
    string caption
)
```

Visual Basic Copy Code

```
Public Function AddItem( _
    item As DiagramItem, _
    caption As String _
) As ItemListView
```

Parameters

item
The DiagramItem instance to add.

caption
The caption to display for the item.

Return Value

See Also

AddItem Method Overload List
ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.ControlLoadedScript Method

Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView ControlLoadedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ControlLoadedScript( _
    value As String _
) As ItemListView
```

Parameters

value
The name of the function

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.FromJson Method

Overload List

Name	Description
ItemListView.FromJson (String)	Creates an ItemListView object from JSON.
ItemListView.FromJson (String, String)	Creates an ItemListView object from JSON.

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.FromJson Method (String)

Creates an ItemListView object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static ItemListView FromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    json As String _
) As ItemListView
```

Parameters

json
The JSON string

Return Value

See Also

FromJson Method Overload List
ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.FromJson Method (String, String)

Creates an ItemListView object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static ItemListView FromJson (
    string id,
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    id As String, _
    json As String _
) As ItemListView
```

Parameters

id
The Id of the ItemListView

json
The JSON string

Return Value

See Also

FromJson Method Overload List
ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.FromRequest Method

Initializes a new instance of the ItemListView class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static ItemListView FromRequest (
    string id,
    HttpRequest request
)
```

Visual Basic Copy Code

```
Public Shared Function FromRequest( _
    id As String, _
    request As HttpRequest _
) As ItemListView
```

Parameters

id
The Id of the ItemListView

request
An HttpRequest object that contains the json data for the view.

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.Init Method

Renders the ItemListView's initializing scripts on a web page under the specified function name.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Init (
    string functionName
)
```

Visual Basic Copy Code

```
Public Function Init( _
    functionName As String _
) As String
```

Parameters

functionName

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.LinkSelectedScript Method

Sets the name of a JavaScript function that should be invoked when a link is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView LinkSelectedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkSelectedScript( _
    value As String _
) As ItemListView
```

Parameters

value
The name of the function

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.LoadFromJson Method

Loads the specified JSON data.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void LoadFromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Sub LoadFromJson( _
    json As String _
)
```

Parameters

json
The JSON string

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.NodeSelectedScript Method

Sets the name of a JavaScript function that should be invoked when a node is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView NodeSelectedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeSelectedScript( _
    value As String _
) As ItemListView
```

Parameters

value
The name of the function

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.RegisterItemType Method

Overload List

Name	Description
ItemListView.RegisterItemType (Type, String, String, Int32)	Registers a custom item class for serialization support.
ItemListView.RegisterItemType (Type, String, String, Int32, String)	Registers a custom item class for serialization support.

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.RegisterItemType Method (Type, String, String, Int32)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView RegisterItemType (
    Type itemClass,
    string classId,
    string clientClassId,
    int classVersion
)
```

Visual Basic Copy Code

```
Public Function RegisterItemType( _
    itemClass As Type, _
    classId As String, _
    clientClassId As String, _
    classVersion As Integer _
) As ItemListView
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when serializing, saving and loading items of the specified type.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

classVersion
A revision number of the item's class serialization format.

Return Value

See Also

RegisterItemType Method Overload List
ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.RegisterItemType Method (Type, String, String, Int32, String)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView RegisterItemType (
    Type itemClass,
    string classId,
    string clientClassId,
    int classVersion,
    string jsonTemplate
)
```

Visual Basic Copy Code

```
Public Function RegisterItemType( _
    itemClass As Type, _
    classId As String, _
    clientClassId As String, _
    classVersion As Integer, _
    jsonTemplate As String _
) As ItemListView
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when serializing, saving and loading items of the specified type.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

classVersion
A revision number of the item's class serialization format.

jsonTemplate
A JSON string representation of the node contents.

Return Value

See Also

RegisterItemType Method Overload List
ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.RegisterSession Method

Registers the session instance with the ItemTypeResolver used in custom item serialization.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView RegisterSession (
    ISession session
)
```

Visual Basic Copy Code

```
Public Function RegisterSession( _
    session As ISession _
) As ItemListView
```

Parameters

session
An ISession instance.

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.RemoveItem Method

Removes a DiagramItem from the displayed items collection.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView RemoveItem (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Function RemoveItem( _
    item As DiagramItem _
) As ItemListView
```

Parameters

item

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.Render Method

Renders the ItemListView on a web page.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Render ()
```

Visual Basic Copy Code

```
Public Function Render () As String
```

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SaveToJson Method

Serializes the ItemListView's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string SaveToJson ()
```

Visual Basic Copy Code

```
Public Function SaveToJson () As String
```

Return Value

The JSON string.

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetAddScriptReferences Method

Sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetAddScriptReferences (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAddScriptReferences( _
    value As Boolean _
) As ItemListView
```

Parameters

value

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetAutoConnectDistance Method

Sets the maximum distance to a node when auto connecting links.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetAutoConnectDistance (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetAutoConnectDistance( _
    value As Single _
) As ItemListView
```

Parameters

value

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetAutoConnectLinks Method

Sets whether links should be automatically connected to the nearest node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetAutoConnectLinks (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAutoConnectLinks( _
    value As Boolean _
) As ItemListView
```

Parameters

value

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetBackColor Method

Sets the background color of the ItemListView control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetBackColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetBackColor( _
    value As Color _
) As ItemListView
```

Parameters

value

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetColumns Method

Sets the number of columns in control's layout.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetColumns (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetColumns( _
    value As Integer _
) As ItemListView
```

Parameters

value

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetEnabled Method

Sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetEnabled (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetEnabled( _
    value As Boolean _
) As ItemListView
```

Parameters

value

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetFont Method

Sets the font of the items captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetFont (
    Font value
)
```

Visual Basic Copy Code

```
Public Function SetFont( _
    value As Font _
) As ItemListView
```

Parameters

value

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetIconSize Method

Sets the size of the items displayed in the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetIconSize (
    SizeF value
)
```

Visual Basic Copy Code

```
Public Function SetIconSize( _
    value As SizeF _
) As ItemListView
```

Parameters

value
The size of the items

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetLabelPosition Method

Sets a value indicating the position of labels relative to item icons.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetLabelPosition (
    LabelPosition value
)
```

Visual Basic Copy Code

```
Public Function SetLabelPosition( _
    value As LabelPosition _
) As ItemListView
```

Parameters

value

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetMinVisibleFontSize Method

Sets a threshold value that hides text if scaled font sizes become smaller.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetMinVisibleFontSize (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetMinVisibleFontSize( _
    value As Single _
) As ItemListView
```

Parameters

value

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetNewInstanceSize Method

Sets the default size with which items dragged onto a diagram will be created.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetNewInstanceSize (
    SizeF value
)
```

Visual Basic Copy Code

```
Public Function SetNewInstanceSize( _
    value As SizeF _
) As ItemListView
```

Parameters

value
The size of the items

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetOrientation Method

Sets the layout orientation of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetOrientation (
    Orientation value
)
```

Visual Basic Copy Code

```
Public Function SetOrientation( _
    value As Orientation _
) As ItemListView
```

Parameters

value

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.SetTextColor Method

Sets the text color of the items captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView SetTextColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetTextColor( _
    value As Color _
) As ItemListView
```

Parameters

value

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.ShapeLibraryLoadedScript Method

Sets the name of a JavaScript function that should be invoked when a shape library is loaded.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListView ShapeLibraryLoadedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ShapeLibraryLoadedScript( _
    value As String _
) As ItemListView
```

Parameters

value
The name of the function

Return Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.ToJson Method

Serializes the ItemListView's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static string ToJson (
    ItemListView view
)
```

Visual Basic Copy Code

```
Public Shared Function ToJson( _
    view As ItemListView _
) As String
```

Parameters

view

Return Value

The JSON string.

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView Properties

ItemListView.AddScriptReferences Property

Gets or sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AddScriptReferences { get; set; }
```

Visual Basic Copy Code

```
Public Property AddScriptReferences As Boolean
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.AutoClasses Property

Gets the registered client types with JSON templates.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Dictionary<string, string> AutoClasses { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property AutoClasses As Dictionary(Of String, String)
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.AutoConnectDistance Property

Gets or sets the maximum distance to a node when auto connecting links.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float AutoConnectDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoConnectDistance As Single
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.AutoConnectLinks Property

Gets or sets whether links should be automatically connected to the nearest node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AutoConnectLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoConnectLinks As Boolean
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.BackColor Property

Gets or sets the background color of the ItemListView control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color BackColor { get; set; }
```

Visual Basic Copy Code

```
Public Property BackColor As Color
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.ClientScriptLocation Property

Gets or sets the URL to the MindFusion.Diagramming.js file.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string ClientScriptLocation { get; set; }
```

Visual Basic Copy Code

```
Public Property ClientScriptLocation As String
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.Columns Property

Gets or sets the number of columns in control's layout.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Columns { get; set; }
```

Visual Basic Copy Code

```
Public Property Columns As Integer
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.Enabled Property

Gets or sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool Enabled { get; set; }
```

Visual Basic Copy Code

```
Public Property Enabled As Boolean
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.Font Property

Gets or sets the font of the items captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Font Font { get; set; }
```

Visual Basic Copy Code

```
Public Property Font As Font
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.IconSize Property

Gets or sets the size of the items displayed in the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public SizeF IconSize { get; set; }
```

Visual Basic Copy Code

```
Public Property IconSize As SizeF
```

Property Value

See Also

ItemListView.Id Property

Gets or sets the id of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.Items Property

Gets or sets the collection of items displayed in the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramItemCollection Items { get; set; }
```

Visual Basic Copy Code

```
Public Property Items As DiagramItemCollection
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.LabelPosition Property

Gets or sets a value indicating the position of labels relative to item icons.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public LabelPosition LabelPosition { get; set; }
```

Visual Basic

```
Public Property LabelPosition As LabelPosition
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.MeasureUnit Property

Gets or sets the unit of measure used to specify item sizes.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public MeasureUnit MeasureUnit { get; set; }
```

Visual Basic

```
Public Property MeasureUnit As MeasureUnit
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.MinVisibleFontSize Property

Gets or sets a threshold value that hides text if scaled font sizes become smaller.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public float MinVisibleFontSize { get; set; }
```

Visual Basic

```
Public Property MinVisibleFontSize As Single
```

Property Value

See Also

ItemListView.NewInstanceSize Property

Gets or sets the default size of items created by drag-and-drop from this list view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public SizeF NewInstanceSize { get; set; }
```

Visual Basic Copy Code

```
Public Property NewInstanceSize As SizeF
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.Orientation Property

Gets or sets the layout orientation of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic Copy Code

```
Public Property Orientation As Orientation
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.Padding Property

Gets or sets the padding space between item's slot bounds and its contents.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Padding { get; set; }
```

Visual Basic Copy Code

```
Public Property Padding As Integer
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.ShapeLibraryLocation Property

The location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string ShapeLibraryLocation { get; set; }
```

Visual Basic Copy Code

```
Public Property ShapeLibraryLocation As String
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListView.TextColor Property

Gets or sets the color of the items captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color TextColor { get; set; }
```

Visual Basic Copy Code

```
Public Property TextColor As Color
```

Property Value

See Also

ItemListView Members
ItemListView Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions Class

Contains extension methods of ItemListView

Syntax

C# Copy Code

```
public sealed class ItemListViewExtensions
```

Visual Basic Copy Code

```
Public NotInheritable Class ItemListViewExtensions
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.ItemListViewExtensions

See Also

ItemListViewExtensions Members
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions Members

Contains extension methods of ItemListView

The following tables list the members exposed by the ItemListViewExtensions type.

Public Methods

	Name	Description
	Generate	
	ItemListView	Overloaded.

See Also

ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions Methods

ItemListViewExtensions.Generate Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Generate (
    ItemListView view
)
```

Visual Basic Copy Code

```
Public Shared Function Generate( _
    view As ItemListView _
) As IHtmlContent
```

Parameters

view

Return Value

See Also

ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method

Overload List

Name	Description
ItemListViewExtensions.ItemListView (IHtmlHelper, ItemListView)	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, ItemListView, IDictionary(String, Object))	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, ItemListView, Object)	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, ItemListView, String)	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, ItemListView, String, IDictionary(String, Object))	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, ItemListView, String, Object)	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, String)	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, String, DiagramItemCollection)	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, String, DiagramItemCollection, IDictionary(String, Object))	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, String, DiagramItemCollection, Object)	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, String, DiagramItemCollection, String)	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, String, IDictionary(String, Object))	Returns an ItemListView by using the specified HTML helper.

ItemListViewExtensions.ItemListView (IHtmlHelper, String, Object)	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, String, String)	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, String, String, IDictionary(String, Object))	Returns an ItemListView by using the specified HTML helper.
ItemListViewExtensions.ItemListView (IHtmlHelper, String, String, Object)	Returns an ItemListView by using the specified HTML helper.

See Also

ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, ItemListView)

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    ItemListView view
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    view As ItemListView _
) As IHtmlContent
```

Parameters

helper
The HTML helper

view
The ItemListView

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, ItemListView, IDictionary(String, Object))

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    ItemListView view,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    view As ItemListView, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

view
The ItemListView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, ItemListView, Object)

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    ItemListView view,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    view As ItemListView, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper

The HTML helper

view
The ItemListView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, ItemListView, String)

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    ItemListView view,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    view As ItemListView, _
    initFunction As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

view
The ItemListView

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, ItemListView, String, IDictionary(String, Object))

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    ItemListView view,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    view As ItemListView, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

view
The ItemListView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, ItemListView, String, Object)

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    ItemListView view,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    view As ItemListView, _
```

```
    initFunction As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

view
The ItemListView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, String)

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    string id
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    id As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ItemListView

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, String, DiagramItemCollection)

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    string id,
    DiagramItemCollection items
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    id As String, _
    items As DiagramItemCollection _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ItemListView

items
The items to be displayed

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, String, DiagramItemCollection, IDictionary(String, Object))

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    string id,
    DiagramItemCollection items,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    id As String, _
    items As DiagramItemCollection, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ItemListView

items
The items to be displayed

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, String, DiagramItemCollection, Object)

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    string id,
    DiagramItemCollection items,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    id As String, _
    items As DiagramItemCollection, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ItemListView

items
The items to be displayed

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, String, DiagramItemCollection, String)

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    string id,
    DiagramItemCollection items,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    id As String, _
    items As DiagramItemCollection, _
    initFunction As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ItemListView

items
The items to be displayed

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, String, IDictionary(String, Object))

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    string id,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    id As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ItemListView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, String, Object)

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    string id,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    id As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id

The Id of the ItemListView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, String, String)

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    string id,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    id As String, _
    initFunction As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ItemListView

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, String, String, IDictionary(String, Object))

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    string id,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    id As String, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ItemListView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewExtensions.ItemListView Method (IHtmlHelper, String, String, Object)

Returns an ItemListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ItemListView (
    IHtmlHelper helper,
    string id,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ItemListView( _
    helper As IHtmlHelper, _
    id As String, _
    initFunction As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ItemListView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ItemListView Method Overload List
ItemListViewExtensions Members
ItemListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewTagHelper Class

Tag helper of ItemListView.

Syntax

C# Copy Code

```
public class ItemListViewTagHelper : TagHelper
```

Visual Basic Copy Code

```
Public Class ItemListViewTagHelper
    Inherits TagHelper
```

Example

This class implements the <mindfusion-itemlistview> tag.

Inheritance Hierarchy

System.Object
  Microsoft.AspNetCore.Razor.TagHelpers.TagHelper
    MindFusion.Diagramming.Mvc.ItemListViewTagHelper

See Also

ItemListViewTagHelper Members
MindFusion.Diagramming.Mvc Namespace

ItemListViewTagHelper Members

Tag helper of ItemListView.

The following tables list the members exposed by the ItemListViewTagHelper type.

Public Constructors

	Name	Description
	ItemListViewTagHelper	

Public Properties

	Name	Description
	Component	
	Id	
	Style	

Public Methods

	Name	Description
	Process	

See Also

ItemListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewTagHelper Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemListViewTagHelper ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ItemListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewTagHelper Methods

ItemListViewTagHelper.Process Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override void Process (
    TagHelperContext context,
    TagHelperOutput output
)
```

Visual Basic

```
Public Overrides Sub Process( _
    context As TagHelperContext, _
    output As TagHelperOutput _
)
```

Parameters

context

output

See Also

ItemListViewTagHelper Members
ItemListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewTagHelper Properties

ItemListViewTagHelper.Component Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public ItemListView Component { get; set; }
```

Visual Basic

```
Public Property Component As ItemListView
```

Property Value

See Also

ItemListViewTagHelper Members
ItemListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewTagHelper.Id Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public string Id { get; set; }
```

Visual Basic

```
Public Property Id As String
```

Property Value

See Also

ItemListViewTagHelper Members
ItemListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ItemListViewTagHelper.Style Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As String
```

Property Value

See Also

ItemListViewTagHelper Members
ItemListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver Class

Provides a type resolver for DiagramItem classes.

Syntax

C# Copy Code

```
public class ItemTypeResolver
```

Visual Basic Copy Code

```
Public Class ItemTypeResolver
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.ItemTypeResolver

See Also

ItemTypeResolver Members
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver Members

Provides a type resolver for DiagramItem classes.

The following tables list the members exposed by the ItemTypeResolver type.

Public Constructors

	Name	Description
	ItemTypeResolver	Overloaded.

Public Properties

	Name	Description
	AutoClasses	Gets the registered client types with JSON templates.
	Converters	Gets the registered custom converters.
	ItemTypes	Gets the registered item types.

Public Methods

	Name	Description
	GetResolver	Overloaded.
	RegisterConverters	Registers converters used for JSON serialization of custom item types.
	RegisterItemType	Overloaded.
	ResolveType	Returns the Type of the specified id.
	ResolveTypeId	Returns the id of the specified Type.
	Unregister	Overloaded.

See Also

ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver Constructor

Overload List

Name	Description
ItemTypeResolver (HttpSessionStateBase)	Creates a new instance of the ItemTypeResolver class.
ItemTypeResolver (ISession)	Creates a new instance of the ItemTypeResolver class.

See Also

ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver Constructor (HttpSessionStateBase)

Creates a new instance of the ItemTypeResolver class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemTypeResolver (
    HttpSessionStateBase session
)
```

Visual Basic Copy Code

```
Public New ( _
    session As HttpSessionStateBase _
)
```

Parameters

session
An HttpSessionStateBase instance.

See Also

ItemTypeResolver Constructor Overload List
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver Constructor (ISession)

Creates a new instance of the ItemTypeResolver class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ItemTypeResolver (
    ISession session
)
```

Visual Basic Copy Code

```
Public New ( _
    session As ISession _
)
```

Parameters

session
An ISession instance.

See Also

ItemTypeResolver Constructor Overload List
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver Methods

ItemTypeResolver.GetResolver Method

Overload List

Name	Description
ItemTypeResolver.GetResolver (HttpSessionStateBase)	Returns the ItemTypeResolver instance registered for the specified session.

ItemTypeResolver.GetResolver (ISession)	Returns the ItemTypeResolver instance registered for the specified session.

See Also

ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.GetResolver Method (HttpSessionStateBase)

Returns the ItemTypeResolver instance registered for the specified session.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static ItemTypeResolver GetResolver (
    HttpSessionStateBase session
)
```

Visual Basic Copy Code

```
Public Shared Function GetResolver( _
    session As HttpSessionStateBase _
) As ItemTypeResolver
```

Parameters

session
An HttpSessionStateBase instance.

Return Value

The ItemTypeResolver for this session.

See Also

GetResolver Method Overload List
ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.GetResolver Method (ISession)

Returns the ItemTypeResolver instance registered for the specified session.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static ItemTypeResolver GetResolver (
    ISession session
)
```

Visual Basic Copy Code

```
Public Shared Function GetResolver( _
    session As ISession _
) As ItemTypeResolver
```

Parameters

session
An ISession instance.

Return Value

See Also

GetResolver Method Overload List
ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.RegisterConverters Method

Registers converters used for JSON serialization of custom item types.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void RegisterConverters (
    JavaScriptConverter[] converters
)
```

Visual Basic Copy Code

```
Public Sub RegisterConverters( _
    converters() As JavaScriptConverter _
)
```

Parameters

converters
The converters to register.

See Also

ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.RegisterItemType Method

Overload List

Name	Description
ItemTypeResolver.RegisterItemType (Type, String)	Registers a custom item class for serialization support.
ItemTypeResolver.RegisterItemType (Type, String, String)	Registers a custom item class for serialization support.
ItemTypeResolver.RegisterItemType (Type, String, String, Int32)	Registers a custom item class for serialization support.
ItemTypeResolver.RegisterItemType (Type, String, String, Int32, String)	Registers a custom item class for serialization support.

See Also

ItemTypeResolver Members
ItemTypeResolver Class

ItemTypeResolver.RegisterItemType Method (Type, String)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void RegisterItemType (
    Type itemClass,
    string clientClassId
)
```

Visual Basic Copy Code

```
Public Sub RegisterItemType( _
    itemClass As Type, _
    clientClassId As String _
)
```

Parameters

itemClass
A Type instance identifying the item class.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

See Also

RegisterItemType Method Overload List
ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.RegisterItemType Method (Type, String, String)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void RegisterItemType (
    Type itemClass,
    string clientClassId,
    string jsonTemplate
)
```

Visual Basic Copy Code

```
Public Sub RegisterItemType( _
    itemClass As Type, _
    clientClassId As String, _
    jsonTemplate As String _
)
```

Parameters

itemClass

A Type instance identifying the item class.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

jsonTemplate
A JSON string representation of the node contents.

See Also

RegisterItemType Method Overload List
ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.RegisterItemType Method (Type, String, String, Int32)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void RegisterItemType (
    Type itemClass,
    string classId,
    string clientClassId,
    int classVersion
)
```

Visual Basic Copy Code

```
Public Sub RegisterItemType( _
    itemClass As Type, _
    classId As String, _
    clientClassId As String, _
    classVersion As Integer _
)
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when serializing, saving and loading items of the specified type.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

classVersion
A revision number of the item's class serialization format.

See Also

RegisterItemType Method Overload List
ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.RegisterItemType Method (Type, String, String, Int32, String)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void RegisterItemType (
    Type itemClass,
    string classId,
    string clientClassId,
    int classVersion,
    string jsonTemplate
)
```

Visual Basic Copy Code

```
Public Sub RegisterItemType( _
    itemClass As Type, _
    classId As String, _
    clientClassId As String, _
    classVersion As Integer, _
    jsonTemplate As String _
)
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when serializing, saving and loading items of the specified type.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

classVersion
A revision number of the item's class serialization format.

jsonTemplate
A JSON string representation of the node contents.

See Also

RegisterItemType Method Overload List
ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.ResolveType Method

Returns the Type of the specified id.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Type ResolveType (
    string id
)
```

Visual Basic Copy Code

```
Public Overrides Function ResolveType( _
    id As String _
) As Type
```

Parameters

id

Return Value

See Also

ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.ResolveTypeId Method

Returns the id of the specified Type.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override string ResolveTypeId (
    Type type
)
```

Visual Basic Copy Code

```
Public Overrides Function ResolveTypeId( _
    type As Type _
) As String
```

Parameters

type

Return Value

See Also

ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.Unregister Method

Overload List

Name	Description
ItemTypeResolver.Unregister (HttpSessionStateBase)	Unregisters the ItemTypeResolver instance registered for the corresponding session.
ItemTypeResolver.Unregister (ISession)	Unregisters the ItemTypeResolver instance registered for the corresponding session.

See Also

ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.Unregister Method (HttpSessionStateBase)

Unregisters the ItemTypeResolver instance registered for the corresponding session.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static void Unregister (
    HttpSessionStateBase session
)
```

Visual Basic Copy Code

```
Public Shared Sub Unregister( _
    session As HttpSessionStateBase _
)
```

Parameters

session
An HttpSessionStateBase instance.

See Also

Unregister Method Overload List
ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.Unregister Method (ISession)

Unregisters the ItemTypeResolver instance registered for the corresponding session.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static void Unregister (
    ISession session
)
```

Visual Basic Copy Code

```
Public Shared Sub Unregister( _
    session As ISession _
)
```

Parameters

session
An ISession instance.

See Also

Unregister Method Overload List
ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver Properties

ItemTypeResolver.AutoClasses Property

Gets the registered client types with JSON templates.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Dictionary<string, string> AutoClasses { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property AutoClasses As Dictionary(Of String, String)
```

Property Value

A Dictionary containing JSON templates as values for class names as keys.

See Also

ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.Converters Property

Gets the registered custom converters.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public JavaScriptConverter[] Converters { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Converters As JavaScriptConverter()
```

Property Value

An array containing converter instances.

See Also

ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

ItemTypeResolver.ItemTypes Property

Gets the registered item types.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Dictionary<Type, string> ItemTypes { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ItemTypes As Dictionary(Of Type, String)
```

Property Value

A dictionary containing JavaScript class names as values for their matching server .NET types as keys.

See Also

ItemTypeResolver Members
ItemTypeResolver Class
MindFusion.Diagramming.Mvc Namespace

LaneGridConverter Class

Serializes DiagramView instances in Canvas and ImageMap mode.

Syntax

C# Copy Code

```
public class LaneGridConverter : JavaScriptConverter
```

Visual Basic Copy Code

```
Public Class LaneGridConverter
    Inherits JavaScriptConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.LaneGridConverter

See Also

LaneGridConverter Members
MindFusion.Diagramming.Mvc Namespace

LaneGridConverter Members

Serializes DiagramView instances in Canvas and ImageMap mode.

The following tables list the members exposed by the LaneGridConverter type.

Public Constructors

	Name	Description
	LaneGridConverter	Creates a new instance of the LaneGridConverter class.

Protected Fields

	Name	Description
	view	A reference to the current **DiagramView** control.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

LaneGridConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneGridConverter Fields

LaneGridConverter.view Field

A reference to the current DiagramView control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
protected DiagramView view
```

Visual Basic Copy Code

```
Protected view As DiagramView
```

See Also

LaneGridConverter Members
LaneGridConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneGridConverter Constructor

Creates a new instance of the LaneGridConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public LaneGridConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

LaneGridConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneGridConverter Methods

LaneGridConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

LaneGridConverter Members
LaneGridConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneGridConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

LaneGridConverter Members
LaneGridConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneGridConverter Properties

LaneGridConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

LaneGridConverter Members
LaneGridConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneHeaderCollectionConverter Class

Serializes HeaderCollection instances in Canvas mode.

Syntax

C# Copy Code

```
public class LaneHeaderCollectionConverter : JavaScriptConverter
```

Visual Basic Copy Code

```
Public Class LaneHeaderCollectionConverter
    Inherits JavaScriptConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.LaneHeaderCollectionConverter

See Also

LaneHeaderCollectionConverter Members
MindFusion.Diagramming.Mvc Namespace

LaneHeaderCollectionConverter Members

Serializes HeaderCollection instances in Canvas mode.

The following tables list the members exposed by the LaneHeaderCollectionConverter type.

Public Constructors

	Name	Description
	LaneHeaderCollectionConverter	

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

LaneHeaderCollectionConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneHeaderCollectionConverter Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public LaneHeaderCollectionConverter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

LaneHeaderCollectionConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneHeaderCollectionConverter Methods

LaneHeaderCollectionConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

LaneHeaderCollectionConverter Members
LaneHeaderCollectionConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneHeaderCollectionConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

LaneHeaderCollectionConverter Members
LaneHeaderCollectionConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneHeaderCollectionConverter Properties

LaneHeaderCollectionConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

LaneHeaderCollectionConverter Members
LaneHeaderCollectionConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneHeaderConverter Class

Serializes Header instances in Canvas mode.

Syntax

C# Copy Code

```
public class LaneHeaderConverter : JavaScriptConverter
```

Visual Basic Copy Code

```
Public Class LaneHeaderConverter
    Inherits JavaScriptConverter
```

Inheritance Hierarchy

System.Object
System.Web.Script.Serialization.JavaScriptConverter
MindFusion.Diagramming.Mvc.LaneHeaderConverter

See Also

LaneHeaderConverter Members
MindFusion.Diagramming.Mvc Namespace

LaneHeaderConverter Members

Serializes Header instances in Canvas mode.

The following tables list the members exposed by the LaneHeaderConverter type.

Public Constructors

	Name	Description
	LaneHeaderConverter	

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

LaneHeaderConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneHeaderConverter Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public LaneHeaderConverter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

LaneHeaderConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneHeaderConverter Methods

LaneHeaderConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

LaneHeaderConverter Members
LaneHeaderConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneHeaderConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

LaneHeaderConverter Members
LaneHeaderConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneHeaderConverter Properties

LaneHeaderConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

LaneHeaderConverter Members
LaneHeaderConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneStyleConverter Class

Serializes Lanes.Style instances in Canvas mode.

Syntax

C# Copy Code

```
public class LaneStyleConverter : JavaScriptConverter
```

Visual Basic Copy Code

```
Public Class LaneStyleConverter
    Inherits JavaScriptConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.LaneStyleConverter

See Also

LaneStyleConverter Members

Serializes Lanes.Style instances in Canvas mode.

The following tables list the members exposed by the LaneStyleConverter type.

Public Constructors

	Name	Description
	LaneStyleConverter	

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

LaneStyleConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneStyleConverter Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public LaneStyleConverter ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

LaneStyleConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneStyleConverter Methods

LaneStyleConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

LaneStyleConverter Members
LaneStyleConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneStyleConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

LaneStyleConverter Members
LaneStyleConverter Class
MindFusion.Diagramming.Mvc Namespace

LaneStyleConverter Properties

LaneStyleConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

LaneStyleConverter Members
LaneStyleConverter Class
MindFusion.Diagramming.Mvc Namespace

NodeListView Class

ListBox control with support for hosting Diagram nodes and dragging them to the **Diagram** control.

Syntax

C# Copy Code

```
public class NodeListView
```

Visual Basic Copy Code

```
Public Class NodeListView
```

Remarks

Obsolete

This component has been superseded by ItemListView, which can also display DiagramLink objects and supports multi-column layouts.

Call the AddNode method to add a node to the view.

Drag-and-drop operations create a clone of the dragged node by calling its clone method. You can override clone to copy property values of custom node subclasses.

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.Mvc.NodeListView

See Also

NodeListView Members

ListBox control with support for hosting Diagram nodes and dragging them to the **Diagram** control.

The following tables list the members exposed by the NodeListView type.

Public Constructors

	Name	Description
	NodeListView	Overloaded. Creates an instance of **NodeListView**.

Public Properties

	Name	Description
	AddScriptReferences	Gets or sets a value indicating whether the control should render references to the client scripts.
	AutoClasses	Gets the registered client types with JSON templates.
	BackColor	Gets or sets the background color of the NodeListView control.
	ClientScriptLocation	Gets or sets the URL to the MindFusion.Diagramming.js file.
	Converters	Gets the registered custom converters.
	DefaultNodeSize	Gets or sets the default size with which nodes dragged onto a diagram will be created.
	Enabled	Gets or sets a value indicating whether handling of mouse events is enabled.
	Font	Gets or sets the font of the nodes captions.
	Height	Gets or sets the height of the view.
	Id	Gets or sets the id of the view.
	Items	Gets or sets the collection of nodes displayed in the view.
	ItemTypes	Gets the registered item types.
	MeasureUnit	Gets or sets the measure unit.
	MinVisibleFontSize	Gets or sets a threshold value that hides text if scaled font sizes become smaller.
	NodeSize	Gets or sets the size of the nodes displayed in the view.
	Orientation	Gets or sets the layout orientation of the control.
	Padding	Gets or sets the padding.
	TextColor	Gets or sets the color of the nodes captions.
	Width	Gets or sets the width of the view.

Public Methods

	Name	Description
	AddNode	Overloaded. Adds a DiagramNode to the displayed nodes collection.
	ControlLoadedScript	Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.

	FromJson	Overloaded.
	FromRequest	Overloaded.
	Init	Renders the NodeListView's initializing scripts on a web page under the specified function name.
	LoadFromJson	Loads the specified JSON data.
	NodeSelectedScript	Sets the name of a JavaScript function that should be invoked when a node is selected, either programmatically or by the user.
	RegisterConverters	Registers custom converters used for custom item types serialization.
	RegisterItemType	Overloaded.
	RegisterSession	Overloaded.
	RemoveNode	Removes a **DiagramNode** from the displayed nodes collection.
	Render	Renders the **NodeListView** on a web page.
	SaveToJson	Serializes the NodeListView's data to JSON.
	SetAddScriptReferences	Sets a value indicating whether the control should render references to the client scripts.
	SetBackColor	Sets the background color of the NodeListView control.
	SetDefaultNodeSize	Sets the default size with which nodes dragged onto a diagram will be created.
	SetEnabled	Sets a value indicating whether handling of mouse events is enabled.
	SetFont	Sets the font of the nodes captions.
	SetHeight	Sets the height of the view.
	SetMinVisibleFontSize	Sets a threshold value that hides text if scaled font sizes become smaller.
	SetNodeSize	Sets the size of the nodes displayed in the view.
	SetOrientation	Sets the layout orientation of the control.
	SetTextColor	Sets the text color of the nodes captions.
	SetWidth	Sets the width of the view.
	ToJson	Serializes the NodeListView's data to JSON.

See Also

NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView Constructor

Creates an instance of NodeListView.

Overload List

Name	Description
NodeListView ()	Creates an instance of NodeListView.
NodeListView (String)	Creates an instance of **NodeListView**.
NodeListView (String, DiagramNodeCollection)	Creates an instance of **NodeListView**.

See Also

NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView Constructor ()

Creates an instance of NodeListView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

NodeListView Constructor Overload List
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView Constructor (String)

Creates an instance of NodeListView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView (
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String _
)
```

Parameters

id
The Id of the NodeListView.

See Also

NodeListView Constructor Overload List
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView Constructor (String, DiagramNodeCollection)

Creates an instance of NodeListView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView (
    string id,
    DiagramNodeCollection items
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String, _
    items As DiagramNodeCollection _
)
```

Parameters

id

The Id of the NodeListView.

items

The collection of nodes to be displayed in the **NodeListView**.

See Also

NodeListView Constructor Overload List
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView Methods

NodeListView.AddNode Method

Adds a DiagramNode to the displayed nodes collection.

Overload List

Name	Description
NodeListView.AddNode (DiagramNode)	Adds a **DiagramNode** to the displayed nodes collection.
NodeListView.AddNode (DiagramNode, String)	Adds a **DiagramNode** to the displayed nodes collection.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.AddNode Method (DiagramNode)

Adds a DiagramNode to the displayed nodes collection.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView AddNode (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Function AddNode( _
    node As DiagramNode _
) As NodeListView
```

Parameters

node
The DiagramNode instance to add.

Return Value

See Also

AddNode Method Overload List
NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.AddNode Method (DiagramNode, String)

Adds a DiagramNode to the displayed nodes collection.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView AddNode (
    DiagramNode node,
    string caption
)
```

Visual Basic Copy Code

```
Public Function AddNode( _
    node As DiagramNode, _
    caption As String _
) As NodeListView
```

Parameters

node
The DiagramNode instance to add.

caption
The caption to display for the node.

Return Value

See Also

AddNode Method Overload List
NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.ControlLoadedScript Method

Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView ControlLoadedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ControlLoadedScript( _
    value As String _
) As NodeListView
```

Parameters

value
The name of the function

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.FromJson Method

Overload List

Name	Description
NodeListView.FromJson (String)	Creates a NodeListView object from JSON.
NodeListView.FromJson (String, String)	Creates a NodeListView object from JSON.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.FromJson Method (String)

Creates a NodeListView object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static NodeListView FromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    json As String _
```

```
) As NodeListView
```

Parameters

json
The JSON string

Return Value

See Also

FromJson Method Overload List
NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.FromJson Method (String, String)

Creates a NodeListView object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static NodeListView FromJson (
    string id,
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    id As String, _
    json As String _
) As NodeListView
```

Parameters

id
The Id of the NodeListView

json
The JSON string

Return Value

See Also

FromJson Method Overload List
NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.FromRequest Method

Overload List

Name	Description
NodeListView.FromRequest (String, HttpRequest)	Creates an instance of NodeListView.
NodeListView.FromRequest (String, HttpRequestBase)	Creates an instance of NodeListView.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.FromRequest Method (String, HttpRequest)

Creates an instance of NodeListView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static NodeListView FromRequest (
    string id,
    HttpRequest request
)
```

Visual Basic Copy Code

```
Public Shared Function FromRequest( _
    id As String, _
    request As HttpRequest _
) As NodeListView
```

Parameters

id
The Id of the NodeListView

request
An HttpRequest object that contains the json data for the view.

Return Value

See Also

FromRequest Method Overload List
NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.FromRequest Method (String, HttpRequestBase)

Creates an instance of NodeListView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static NodeListView FromRequest (
    string id,
    HttpRequestBase request
)
```

Visual Basic Copy Code

```
Public Shared Function FromRequest( _
    id As String, _
```

```
    request As HttpRequestBase _
) As NodeListView
```

Parameters

id
The Id of the NodeListView

request
An HttpRequest object that contains the json data for the view.

Return Value

See Also

FromRequest Method Overload List
NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.Init Method

Renders the NodeListView's initializing scripts on a web page under the specified function name.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Init (
    string functionName
)
```

Visual Basic Copy Code

```
Public Function Init( _
    functionName As String _
) As String
```

Parameters

functionName

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.LoadFromJson Method

Loads the specified JSON data.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void LoadFromJson (
    string json
)
```

Visual Basic

```
Public Sub LoadFromJson( _
    json As String _
)
```

Parameters

json
The JSON string

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.NodeSelectedScript Method

Sets the name of a JavaScript function that should be invoked when a node is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public NodeListView NodeSelectedScript (
    string value
)
```

Visual Basic

```
Public Function NodeSelectedScript( _
    value As String _
) As NodeListView
```

Parameters

value
The name of the function

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.RegisterConverters Method

Registers custom converters used for custom item types serialization.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public NodeListView RegisterConverters (
    JavaScriptConverter[] converters
)
```

Visual Basic

```
Public Function RegisterConverters( _
    converters() As JavaScriptConverter _
) As NodeListView
```

Parameters

converters
The converters to register.

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.RegisterItemType Method

Overload List

Name	Description
NodeListView.RegisterItemType (Type, String, String, Int32)	Registers a custom item class for serialization support.
NodeListView.RegisterItemType (Type, String, String, Int32, String)	Registers a custom item class for serialization support.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.RegisterItemType Method (Type, String, String, Int32)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView RegisterItemType (
    Type itemClass,
    string classId,
    string clientClassId,
    int classVersion
)
```

Visual Basic Copy Code

```
Public Function RegisterItemType( _
    itemClass As Type, _
    classId As String, _
    clientClassId As String, _
    classVersion As Integer _
) As NodeListView
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when serializing, saving and loading items of the specified type.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

classVersion
A revision number of the item's class serialization format.

Return Value

See Also

RegisterItemType Method Overload List
NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.RegisterItemType Method (Type, String, String, Int32, String)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView RegisterItemType (
    Type itemClass,
    string classId,
    string clientClassId,
    int classVersion,
    string jsonTemplate
)
```

Visual Basic Copy Code

```
Public Function RegisterItemType( _
    itemClass As Type, _
    classId As String, _
    clientClassId As String, _
    classVersion As Integer, _
    jsonTemplate As String _
) As NodeListView
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when serializing, saving and loading items of the specified type.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

classVersion
A revision number of the item's class serialization format.

jsonTemplate
A JSON string representation of the node contents.

Return Value

See Also

RegisterItemType Method Overload List
NodeListView Members

NodeListView.RegisterSession Method

Overload List

Name	Description
NodeListView.RegisterSession (HttpSessionStateBase)	Registers the session instance with the ItemTypeResolver used in custom item serialization.
NodeListView.RegisterSession (ISession)	Registers the session instance with the ItemTypeResolver used in custom item serialization.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.RegisterSession Method (HttpSessionStateBase)

Registers the session instance with the ItemTypeResolver used in custom item serialization.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView RegisterSession (
    HttpSessionStateBase session
)
```

Visual Basic Copy Code

```
Public Function RegisterSession( _
    session As HttpSessionStateBase _
) As NodeListView
```

Parameters

session
An HttpSessionStateBase instance.

Return Value

This NodeListView instance for method chaining.

See Also

RegisterSession Method Overload List
NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.RegisterSession Method (ISession)

Registers the session instance with the ItemTypeResolver used in custom item serialization.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView RegisterSession (
    ISession session
)
```

Visual Basic Copy Code

```
Public Function RegisterSession( _
    session As ISession _
) As NodeListView
```

Parameters

session
An ISession instance.

Return Value

See Also

RegisterSession Method Overload List
NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.RemoveNode Method

Removes a DiagramNode from the displayed nodes collection.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView RemoveNode (
    DiagramNode node
)
```

Visual Basic Copy Code

```
Public Function RemoveNode( _
    node As DiagramNode _
) As NodeListView
```

Parameters

node
The DiagramNode instance to remove.

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.Render Method

Renders the NodeListView on a web page.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Render ()
```

Visual Basic Copy Code

```
Public Function Render () As String
```

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SaveToJson Method

Serializes the NodeListView's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string SaveToJson ()
```

Visual Basic Copy Code

```
Public Function SaveToJson () As String
```

Return Value

The JSON string.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SetAddScriptReferences Method

Sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView SetAddScriptReferences (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAddScriptReferences( _
    value As Boolean _
) As NodeListView
```

Parameters

value

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SetBackColor Method

Sets the background color of the NodeListView control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView SetBackColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetBackColor( _
    value As Color _
) As NodeListView
```

Parameters

value

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SetDefaultNodeSize Method

Sets the default size with which nodes dragged onto a diagram will be created.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView SetDefaultNodeSize (
    Size value
)
```

Visual Basic Copy Code

```
Public Function SetDefaultNodeSize( _
    value As Size _
) As NodeListView
```

Parameters

value
The size of the nodes

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SetEnabled Method

Sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView SetEnabled (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetEnabled( _
    value As Boolean _
) As NodeListView
```

Parameters

value

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SetFont Method

Sets the font of the nodes captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView SetFont (
    Font value
)
```

Visual Basic Copy Code

```
Public Function SetFont( _
    value As Font _
) As NodeListView
```

Parameters

value

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SetHeight Method

Sets the height of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView SetHeight (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetHeight( _
    value As Integer _
) As NodeListView
```

Parameters

value
The height of the view in pixels

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SetMinVisibleFontSize Method

Sets a threshold value that hides text if scaled font sizes become smaller.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView SetMinVisibleFontSize (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetMinVisibleFontSize( _
    value As Single _
) As NodeListView
```

Parameters

value
A **float** value specifying the smallest visible font size.

Return Value

This NodeListView instance for chaining method calls.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SetNodeSize Method

Sets the size of the nodes displayed in the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView SetNodeSize (
    Size value
)
```

Visual Basic Copy Code

```
Public Function SetNodeSize( _
    value As Size _
) As NodeListView
```

Parameters

value
The size of the nodes

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SetOrientation Method

Sets the layout orientation of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView SetOrientation (
    Orientation value
)
```

Visual Basic

```
Public Function SetOrientation( _
    value As Orientation _
) As NodeListView
```

Parameters

value

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SetTextColor Method

Sets the text color of the nodes captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public NodeListView SetTextColor (
    Color value
)
```

Visual Basic

```
Public Function SetTextColor( _
    value As Color _
) As NodeListView
```

Parameters

value

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.SetWidth Method

Sets the width of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public NodeListView SetWidth (
    int value
)
```

Visual Basic

```
Public Function SetWidth( _
    value As Integer _
) As NodeListView
```

Parameters

value
The width of the view in pixels

Return Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.ToJson Method

Serializes the NodeListView's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static string ToJson (
    NodeListView view
)
```

Visual Basic Copy Code

```
Public Shared Function ToJson( _
    view As NodeListView _
) As String
```

Parameters

view

Return Value

The JSON string.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView Properties

NodeListView.AddScriptReferences Property

Gets or sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AddScriptReferences { get; set; }
```

Visual Basic

```
Public Property AddScriptReferences As Boolean
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.AutoClasses Property

Gets the registered client types with JSON templates.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public Dictionary<string, string> AutoClasses { get; }
```

Visual Basic

```
Public ReadOnly Property AutoClasses As Dictionary(Of String, String)
```

Property Value

A **Dictionary** containing registered JSON templates for automatic client-side node types.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.BackColor Property

Gets or sets the background color of the NodeListView control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public Color BackColor { get; set; }
```

Visual Basic

```
Public Property BackColor As Color
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.ClientScriptLocation Property

Gets or sets the URL to the MindFusion.Diagramming.js file.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string ClientScriptLocation { get; set; }
```

Visual Basic Copy Code

```
Public Property ClientScriptLocation As String
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.Converters Property

Gets the registered custom converters.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public JavaScriptConverter[] Converters { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Converters As JavaScriptConverter()
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.DefaultNodeSize Property

Gets or sets the default size with which nodes dragged onto a diagram will be created.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Size DefaultNodeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property DefaultNodeSize As Size
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.Enabled Property

Gets or sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool Enabled { get; set; }
```

Visual Basic Copy Code

```
Public Property Enabled As Boolean
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.Font Property

Gets or sets the font of the nodes captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Font Font { get; set; }
```

Visual Basic Copy Code

```
Public Property Font As Font
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.Height Property

Gets or sets the height of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Height { get; set; }
```

Visual Basic Copy Code

```
Public Property Height As Integer
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.Id Property

Gets or sets the id of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.Items Property

Gets or sets the collection of nodes displayed in the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramNodeCollection Items { get; set; }
```

Visual Basic Copy Code

```
Public Property Items As DiagramNodeCollection
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.ItemTypes Property

Gets the registered item types.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Dictionary<Type, string> ItemTypes { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property ItemTypes As Dictionary(Of Type, String)
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.MeasureUnit Property

Gets or sets the measure unit.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public MeasureUnit MeasureUnit { get; set; }
```

Visual Basic Copy Code

```
Public Property MeasureUnit As MeasureUnit
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.MinVisibleFontSize Property

Gets or sets a threshold value that hides text if scaled font sizes become smaller.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float MinVisibleFontSize { get; set; }
```

Visual Basic Copy Code

```
Public Property MinVisibleFontSize As Single
```

Property Value

A **float** value specifying the smallest visible font size.

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.NodeSize Property

Gets or sets the size of the nodes displayed in the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Size NodeSize { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeSize As Size
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.Orientation Property

Gets or sets the layout orientation of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic Copy Code

```
Public Property Orientation As Orientation
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.Padding Property

Gets or sets the padding.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Padding { get; set; }
```

Visual Basic Copy Code

```
Public Property Padding As Integer
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.TextColor Property

Gets or sets the color of the nodes captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color TextColor { get; set; }
```

Visual Basic Copy Code

```
Public Property TextColor As Color
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListView.Width Property

Gets or sets the width of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Width { get; set; }
```

Visual Basic Copy Code

```
Public Property Width As Integer
```

Property Value

See Also

NodeListView Members
NodeListView Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions Class

Contains extension methods of NodeListView

Syntax

C# Copy Code

```
public sealed class NodeListViewExtensions
```

Visual Basic Copy Code

```
Public NotInheritable Class NodeListViewExtensions
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.NodeListViewExtensions

See Also

NodeListViewExtensions Members
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions Members

Contains extension methods of NodeListView

The following tables list the members exposed by the NodeListViewExtensions type.

Public Methods

	Name	Description
	NodeListView	Overloaded. Returns a **NodeListView** by using the specified HTML helper.

See Also

NodeListViewExtensions Methods

NodeListViewExtensions.NodeListView Method

Returns a NodeListView by using the specified HTML helper.

Overload List

Name	Description
NodeListViewExtensions.NodeListView (IHtmlHelper, NodeListView)	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, NodeListView, IDictionary(String, Object))	Returns a NodeListView by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, NodeListView, Object)	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, NodeListView, String)	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, NodeListView, String, IDictionary(String, Object))	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, NodeListView, String, Object)	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, String)	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, String, DiagramNodeCollection)	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, String, DiagramNodeCollection, IDictionary(String, Object))	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, String, DiagramNodeCollection, Object)	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, String, DiagramNodeCollection, String)	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, String, IDictionary(String, Object))	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, String, Object)	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, String, String)	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, String, String, IDictionary(String, Object))	Returns a **NodeListView** by using the specified HTML helper.
NodeListViewExtensions.NodeListView (IHtmlHelper, String, String, Object)	Returns a **NodeListView** by using the specified HTML helper.

See Also

NodeListViewExtensions.NodeListView Method (IHtmlHelper, NodeListView)

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString NodeListView (
    HtmlHelper helper,
    NodeListView view
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As HtmlHelper, _
    view As NodeListView _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The NodeListView.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, NodeListView, IDictionary(String, Object))

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString NodeListView (
    HtmlHelper helper,
    NodeListView view,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As HtmlHelper, _
    view As NodeListView, _
    htmlAttributes As IDictionary(Of String, Object) _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The NodeListView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, NodeListView, Object)

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString NodeListView (
    HtmlHelper helper,
    NodeListView view,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As HtmlHelper, _
    view As NodeListView, _
    htmlAttributes As Object _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The NodeListView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members

NodeListViewExtensions.NodeListView Method (IHtmlHelper, NodeListView, String)

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString NodeListView (
    HtmlHelper helper,
    NodeListView view,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As HtmlHelper, _
    view As NodeListView, _
    initFunction As String _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The NodeListView

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, NodeListView, String, IDictionary(String, Object))

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString NodeListView (
    HtmlHelper helper,
    NodeListView view,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As HtmlHelper, _
    view As NodeListView, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The NodeListView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, NodeListView, String, Object)

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString NodeListView (
    HtmlHelper helper,
    NodeListView view,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As HtmlHelper, _
    view As NodeListView, _
    initFunction As String, _
    htmlAttributes As Object _
) As MvcHtmlString
```

Parameters

helper

The HTML helper

view
The NodeListView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, String)

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString NodeListView (
    HtmlHelper helper,
    string id
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As IHtmlHelper, _
    id As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the NodeListView.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, String, DiagramNodeCollection)

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString NodeListView (
    HtmlHelper helper,
    string id,
    DiagramNodeCollection items
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As HtmlHelper, _
    id As String, _
    items As DiagramNodeCollection _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

id
The Id of the NodeListView.

items
The items to be displayed

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, String, DiagramNodeCollection, IDictionary(String, Object))

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString NodeListView (
    HtmlHelper helper,
    string id,
    DiagramNodeCollection items,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As HtmlHelper, _
    id As String, _
    items As DiagramNodeCollection, _
    htmlAttributes As IDictionary(Of String, Object) _
) As MvcHtmlString
```

Parameters

helper

The HTML helper

id
The Id of the NodeListView

items
The items to be displayed

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, String, DiagramNodeCollection, Object)

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString NodeListView (
    HtmlHelper helper,
    string id,
    DiagramNodeCollection items,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As HtmlHelper, _
    id As String, _
    items As DiagramNodeCollection, _
    htmlAttributes As Object _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

id
The Id of the NodeListView

items
The items to be displayed

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, String, DiagramNodeCollection, String)

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString NodeListView (
    HtmlHelper helper,
    string id,
    DiagramNodeCollection items,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As HtmlHelper, _
    id As String, _
    items As DiagramNodeCollection, _
    initFunction As String _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

id
The Id of the NodeListView

items
The items to be displayed

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, String, IDictionary(String, Object))

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent NodeListView (
    IHtmlHelper helper,
    string id,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As IHtmlHelper, _
    id As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the NodeListView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, String, Object)

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent NodeListView (
    IHtmlHelper helper,
    string id,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As IHtmlHelper, _
    id As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the NodeListView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members

NodeListViewExtensions.NodeListView Method (IHtmlHelper, String, String)

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent NodeListView (
    IHtmlHelper helper,
    string id,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As IHtmlHelper, _
    id As String, _
    initFunction As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the NodeListView

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, String, String, IDictionary(String, Object))

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent NodeListView (
    IHtmlHelper helper,
    string id,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As IHtmlHelper, _
    id As String, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the NodeListView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewExtensions.NodeListView Method (IHtmlHelper, String, String, Object)

Returns a NodeListView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent NodeListView (
    IHtmlHelper helper,
    string id,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function NodeListView( _
    helper As IHtmlHelper, _
    id As String, _
    initFunction As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper

The HTML helper

id
The Id of the NodeListView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

NodeListView Method Overload List
NodeListViewExtensions Members
NodeListViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewTagHelper Class

Tag helper of NodeListView.

Syntax

C# Copy Code

```
public class NodeListViewTagHelper : TagHelper
```

Visual Basic Copy Code

```
Public Class NodeListViewTagHelper
    Inherits TagHelper
```

Remarks

This class implements the <mindfusion-nodelistview> tag.

Inheritance Hierarchy

System.Object
  Microsoft.AspNetCore.Razor.TagHelpers.TagHelper
    MindFusion.Diagramming.Mvc.NodeListViewTagHelper

See Also

NodeListViewTagHelper Members
MindFusion.Diagramming.Mvc Namespace

NodeListViewTagHelper Members

Tag helper of NodeListView.

The following tables list the members exposed by the NodeListViewTagHelper type.

Public Constructors

	Name	Description
	NodeListViewTagHelper	

Public Properties

	Name	Description
	Component	

	Id	
	Style	

Public Methods

	Name	Description
	Process	

See Also

NodeListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewTagHelper Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListViewTagHelper ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

NodeListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewTagHelper Methods

NodeListViewTagHelper.Process Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override void Process (
    TagHelperContext context,
    TagHelperOutput output
)
```

Visual Basic Copy Code

```
Public Overrides Sub Process( _
    context As TagHelperContext, _
    output As TagHelperOutput _
)
```

Parameters

context

output

See Also

NodeListViewTagHelper Members
NodeListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewTagHelper Properties

NodeListViewTagHelper.Component Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public NodeListView Component { get; set; }
```

Visual Basic Copy Code

```
Public Property Component As NodeListView
```

Property Value

See Also

NodeListViewTagHelper Members
NodeListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewTagHelper.Id Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

NodeListViewTagHelper Members
NodeListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

NodeListViewTagHelper.Style Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As String
```

Property Value

See Also

NodeListViewTagHelper Members
NodeListViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

Orientation Enumeration

Specifies a general orientation.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public enum Orientation
```

Visual Basic Copy Code

```
Public Enum Orientation
```

Members

	Member name	Description
	Horizontal	Indicates horizontal orientation.
	Vertical	Indicates vertical orientation.

See Also

MindFusion.Diagramming.Mvc Namespace

Overview Class

Provides a scaled-down view of a DiagramView contents and lets the user scroll the bound view.

Syntax

C# Copy Code

```
public class Overview
```

Visual Basic Copy Code

```
Public Class Overview
```

Inheritance Hierarchy

System.Object
MindFusion.Diagramming.Mvc.Overview

See Also

Overview Members
MindFusion.Diagramming.Mvc Namespace

Overview Members

Provides a scaled-down view of a DiagramView contents and lets the user scroll the bound view.

The following tables list the members exposed by the Overview type.

Public Constructors

	Name	Description
	Overview	Overloaded.

Public Properties

	Name	Description
	AddScriptReferences	Gets or sets a value indicating whether the control should render references to the client scripts.
	AllowZoom	Gets or sets a value indicating whether users are allowed to zoom the diagram by resizing the overview's viewport tracker rectangle.
	AutoScrollAmount	Gets or sets the amount by which to auto-scroll the view when the mouse leaves the view boundaries while drawing.
	BackColor	Gets or sets the background color of the overview control.
	ClientScriptLocation	Gets or sets the URL to the MindFusion.Diagramming.js file.
	DiagramView	Gets or sets the id of the **DiagramView**.
	DimColor	Gets or sets a color of the displayed diagram.
	Enabled	Gets or sets a value indicating whether handling of mouse events is enabled.
	Height	Gets or sets the height of the view.
	Id	Gets or sets the id of the view.
	MinVisibleFontSize	Gets or sets a threshold value that hides text if scaled font sizes become smaller.
	ScaleFactor	Gets or sets a fixed scale factor to be used when FitAll is disabled.
	ScaleMode	Gets or sets a value indicating whether to fit the whole diagram inside the overview control.
	Width	Gets or sets the width of the view.

Public Methods

	Name	Description
	ControlLoadedScript	Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.
	FromJson	Overloaded.

	FromRequest	Overloaded.
	Init	Renders the Overview's initializing scripts on a web page under the specified function name.
	LoadFromJson	Loads the specified JSON data.
	Render	Renders the **Overview** on a web page.
	SaveToJson	Serializes the Overview's data to JSON.
	SetAddScriptReferences	Sets a value indicating whether the control should render references to the client scripts.
	SetAllowZoom	Sets a value indicating whether users are allowed to zoom the diagram by resizing the overview's viewport tracker rectangle.
	SetAutoScrollAmount	Sets the amount by which to auto-scroll the view when the mouse leaves the view boundaries while drawing.
	SetBackColor	Sets the background color of the overview control.
	SetDiagramView	Sets the id of the DiagramView.
	SetDimColor	Sets a color of the displayed diagram.
	SetEnabled	Sets a value indicating whether handling of mouse events is enabled.
	SetHeight	Sets the height of the view.
	SetMinVisibleFontSize	Sets a threshold value that hides text if scaled font sizes become smaller.
	SetScaleFactor	Sets a fixed scale factor to be used when **FitAll** is disabled.
	SetScaleMode	Sets a value indicating whether to fit the whole diagram inside the overview control.
	SetWidth	Sets the width of the view.
	ToJson	Serializes the Overview's data to JSON.

See Also

Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview Constructor

Overload List

Name	Description
Overview ()	Creates an instance of Overview.
Overview (String)	Creates an instance of Overview.
Overview (String, String)	Creates an instance of Overview for a DiagramView.

See Also

Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview Constructor ()

Creates an instance of Overview.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

Overview Constructor Overload List
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview Constructor (String)

Creates an instance of Overview.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview (
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String _
)
```

Parameters

id

See Also

Overview Constructor Overload List
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview Constructor (String, String)

Creates an instance of Overview for a DiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview (
    string id,
    string diagramViewId
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String, _
    diagramViewId As String _
)
```

Parameters

id
The Id of the Overview.

diagramViewId
The Id of the DiagramView

See Also

Overview Constructor Overload List
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview Methods

Overview.ControlLoadedScript Method

Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview ControlLoadedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ControlLoadedScript( _
    value As String _
) As Overview
```

Parameters

value
The name of the function

Return Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.FromJson Method

Overload List

Name	Description
Overview.FromJson (String)	Creates a Overview object from JSON.

Overview.FromJson (String, String)	Creates a Overview object from JSON.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.FromJson Method (String)

Creates a Overview object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static Overview FromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    json As String _
) As Overview
```

Parameters

json
The JSON string

Return Value

See Also

FromJson Method Overload List
Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.FromJson Method (String, String)

Creates a Overview object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static Overview FromJson (
    string id,
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    id As String, _
    json As String _
) As Overview
```

Parameters

id
The Id of the Overview

json
The JSON string

Return Value

See Also

FromJson Method Overload List
Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.FromRequest Method

Overload List

Name	Description
Overview.FromRequest (String, HttpRequest)	Creates an instance of Overview.
Overview.FromRequest (String, HttpRequestBase)	Creates an instance of Overview.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.FromRequest Method (String, HttpRequest)

Creates an instance of Overview.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static Overview FromRequest (
    string id,
    HttpRequest request
)
```

Visual Basic Copy Code

```
Public Shared Function FromRequest( _
    id As String, _
    request As HttpRequest _
) As Overview
```

Parameters

id
The Id of the Overview

request
An HttpRequest object that contains the json data for the view.

Return Value

See Also

FromRequest Method Overload List
Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.FromRequest Method (String, HttpRequestBase)

Creates an instance of Overview.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static Overview FromRequest (
    string id,
    HttpRequestBase request
)
```

Visual Basic Copy Code

```
Public Shared Function FromRequest( _
    id As String, _
    request As HttpRequestBase _
) As Overview
```

Parameters

id
The Id of the Overview

request
An HttpRequest object that contains the json data for the view.

Return Value

See Also

FromRequest Method Overload List
Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.Init Method

Renders the Overview's initializing scripts on a web page under the specified function name.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Init (
    string functionName
)
```

Visual Basic Copy Code

```
Public Function Init( _
    functionName As String _
) As String
```

Parameters

functionName

Return Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.LoadFromJson Method

Loads the specified JSON data.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void LoadFromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Sub LoadFromJson( _
    json As String _
)
```

Parameters

json
The JSON string

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.Render Method

Renders the Overview on a web page.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Render ()
```

Visual Basic Copy Code

```
Public Function Render () As String
```

Return Value

See Also

Overview Members
Overview Class

Overview.SaveToJson Method

Serializes the Overview's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string SaveToJson ()
```

Visual Basic Copy Code

```
Public Function SaveToJson () As String
```

Return Value

The JSON string.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetAddScriptReferences Method

Sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetAddScriptReferences (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAddScriptReferences( _
    value As Boolean _
) As Overview
```

Parameters

value

Return Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetAllowZoom Method

Sets a value indicating whether users are allowed to zoom the diagram by resizing the overview's viewport tracker rectangle.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetAllowZoom (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAllowZoom( _
    value As Boolean _
) As Overview
```

Parameters

value

true to let users zoom by resizing the tracker rectangle, or **false** otherwise.

Return Value

This Overview instance for chaining method calls.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetAutoScrollAmount Method

Sets the amount by which to auto-scroll the view when the mouse leaves the view boundaries while drawing.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetAutoScrollAmount (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetAutoScrollAmount( _
    value As Integer _
) As Overview
```

Parameters

value

A number specifying the amount by which to auto-scroll.

Return Value

This Overview instance for chaining method calls.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetBackColor Method

Sets the background color of the overview control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetBackColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetBackColor( _
    value As Color _
) As Overview
```

Parameters

value

Return Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetDiagramView Method

Sets the id of the DiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetDiagramView (
    string value
)
```

Visual Basic Copy Code

```
Public Function SetDiagramView( _
    value As String _
) As Overview
```

Parameters

value

Return Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetDimColor Method

Sets a color of the displayed diagram.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetDimColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetDimColor( _
    value As Color _
) As Overview
```

Parameters

value

Return Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetEnabled Method

Sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetEnabled (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetEnabled( _
    value As Boolean _
) As Overview
```

Parameters

value

Return Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetHeight Method

Sets the height of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetHeight (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetHeight( _
    value As Integer _
) As Overview
```

Parameters

value
The height of the view in pixels

Return Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetMinVisibleFontSize Method

Sets a threshold value that hides text if scaled font sizes become smaller.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetMinVisibleFontSize (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetMinVisibleFontSize( _
    value As Single _
) As Overview
```

Parameters

value
A **float** value specifying the smallest visible font size.

Return Value

This Overview instance for chaining method calls.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetScaleFactor Method

Sets a fixed scale factor to be used when FitAll is disabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetScaleFactor (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetScaleFactor( _
    value As Single _
) As Overview
```

Parameters

value
A **float** value specifying the scale as percent of actual output. The default is 30.

Return Value

This Overview instance for chaining method calls.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetScaleMode Method

Sets a value indicating whether to fit the whole diagram inside the overview control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetScaleMode (
    ScaleMode value
)
```

Visual Basic Copy Code

```
Public Function SetScaleMode( _
    value As ScaleMode _
) As Overview
```

Parameters

value
One of the ScaleMode enumeration elements. The default value is **FitAll**.

Return Value

This Overview instance for chaining method calls.

Remarks

If fit all mode is enabled, the ScaleFactor is changed to a suitable value small enough to make the diagram's contents entirely visible.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.SetWidth Method

Sets the width of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview SetWidth (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetWidth( _
    value As Integer _
) As Overview
```

Parameters

value
The width of the view in pixels

Return Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.ToJson Method

Serializes the Overview's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static string ToJson (
    Overview view
)
```

Visual Basic Copy Code

```
Public Shared Function ToJson( _
    view As Overview _
) As String
```

Parameters

view

Return Value

The JSON string.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview Properties

Overview.AddScriptReferences Property

Gets or sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AddScriptReferences { get; set; }
```

Visual Basic Copy Code

```
Public Property AddScriptReferences As Boolean
```

Property Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.AllowZoom Property

Gets or sets a value indicating whether users are allowed to zoom the diagram by resizing the overview's viewport tracker rectangle.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AllowZoom { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowZoom As Boolean
```

Property Value

true to let users zoom by resizing the tracker rectangle, or **false** otherwise.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.AutoScrollAmount Property

Gets or sets the amount by which to auto-scroll the view when the mouse leaves the view boundaries while drawing.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float AutoScrollAmount { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoScrollAmount As Single
```

Property Value

A number specifying the amount by which to auto-scroll.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.BackColor Property

Gets or sets the background color of the overview control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color BackColor { get; set; }
```

Visual Basic Copy Code

```
Public Property BackColor As Color
```

Property Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.ClientScriptLocation Property

Gets or sets the URL to the MindFusion.Diagramming.js file.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string ClientScriptLocation { get; set; }
```

Visual Basic Copy Code

```
Public Property ClientScriptLocation As String
```

Property Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.DiagramView Property

Gets or sets the id of the DiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string DiagramView { get; set; }
```

Visual Basic Copy Code

```
Public Property DiagramView As String
```

Property Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.DimColor Property

Gets or sets a color of the displayed diagram.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Brush DimColor { get; set; }
```

Visual Basic Copy Code

```
Public Property DimColor As Brush
```

Property Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.Enabled Property

Gets or sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool Enabled { get; set; }
```

Visual Basic Copy Code

```
Public Property Enabled As Boolean
```

Property Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.Height Property

Gets or sets the height of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Height { get; set; }
```

Visual Basic Copy Code

```
Public Property Height As Integer
```

Property Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.Id Property

Gets or sets the id of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.MinVisibleFontSize Property

Gets or sets a threshold value that hides text if scaled font sizes become smaller.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float MinVisibleFontSize { get; set; }
```

Visual Basic Copy Code

```
Public Property MinVisibleFontSize As Single
```

Property Value

A float value specifying the smallest visible font size.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.ScaleFactor Property

Gets or sets a fixed scale factor to be used when FitAll is disabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float ScaleFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property ScaleFactor As Single
```

Property Value

A **float** value specifying the scale as percent of actual output. The default is 30.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.ScaleMode Property

Gets or sets a value indicating whether to fit the whole diagram inside the overview control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ScaleMode ScaleMode { get; set; }
```

Visual Basic Copy Code

```
Public Property ScaleMode As ScaleMode
```

Property Value

One of the **ScaleMode** enumeration elements. The default value is FitAll.

Remarks

If fit all mode is enabled, the ScaleFactor is changed to a suitable value small enough to make the diagram's contents entirely visible.

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

Overview.Width Property

Gets or sets the width of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Width { get; set; }
```

Visual Basic Copy Code

```
Public Property Width As Integer
```

Property Value

See Also

Overview Members
Overview Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions Class

Contains extension methods of Overview.

Syntax

C# Copy Code

```
public sealed class OverviewExtensions
```

Visual Basic Copy Code

```
Public NotInheritable Class OverviewExtensions
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.OverviewExtensions

See Also

OverviewExtensions Members
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions Members

Contains extension methods of Overview.

The following tables list the members exposed by the OverviewExtensions type.

Public Methods

	Name	Description
	Overview	Overloaded. Returns a **Overview** by using the specified HTML helper.

See Also

OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions Methods

OverviewExtensions.Overview Method

Returns a Overview by using the specified HTML helper.

Overload List

Name	Description
OverviewExtensions.Overview (IHtmlHelper, Overview)	Returns a **Overview** by using the specified HTML helper.
OverviewExtensions.Overview (IHtmlHelper, Overview, IDictionary(String, Object))	Returns a Overview by using the specified HTML helper.
OverviewExtensions.Overview (IHtmlHelper, Overview, Object)	Returns a **Overview** by using the specified HTML helper.
OverviewExtensions.Overview (IHtmlHelper, Overview, String)	Returns a **Overview** by using the specified HTML helper.
OverviewExtensions.Overview (IHtmlHelper, Overview, String, IDictionary(String, Object))	Returns a **Overview** by using the specified HTML helper.
OverviewExtensions.Overview (IHtmlHelper, Overview, String, Object)	Returns a **Overview** by using the specified HTML helper.
OverviewExtensions.Overview (IHtmlHelper, String, String)	Returns a **Overview** by using the specified HTML helper.

OverviewExtensions.Overview (IHtmlHelper, String, String, IDictionary(String, Object))	Returns a **Overview** by using the specified HTML helper.
OverviewExtensions.Overview (IHtmlHelper, String, String, Object)	Returns a **Overview** by using the specified HTML helper.
OverviewExtensions.Overview (IHtmlHelper, String, String, String)	Returns a **Overview** by using the specified HTML helper.
OverviewExtensions.Overview (IHtmlHelper, String, String, String, IDictionary(String, Object))	Returns a **Overview** by using the specified HTML helper.
OverviewExtensions.Overview (IHtmlHelper, String, String, String, Object)	Returns a **Overview** by using the specified HTML helper.

See Also

OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions.Overview Method (IHtmlHelper, Overview)

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Overview (
    HtmlHelper helper,
    Overview view
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As HtmlHelper, _
    view As Overview _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The Overview.

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions.Overview Method (IHtmlHelper, Overview, IDictionary(String, Object))

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Overview (
    HtmlHelper helper,
    Overview view,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As HtmlHelper, _
    view As Overview, _
    htmlAttributes As IDictionary(Of String, Object) _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The Overview

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions.Overview Method (IHtmlHelper, Overview, Object)

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Overview (
    HtmlHelper helper,
    Overview view,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As HtmlHelper, _
    view As Overview, _
    htmlAttributes As Object _
) As MvcHtmlString
```

Parameters

helper

The HTML helper

view
The Overview

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions.Overview Method (IHtmlHelper, Overview, String)

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Overview (
    HtmlHelper helper,
    Overview view,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As HtmlHelper, _
    view As Overview, _
    initFunction As String _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The Overview

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions.Overview Method (IHtmlHelper, Overview, String, IDictionary(String, Object))

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Overview (
    HtmlHelper helper,
    Overview view,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As HtmlHelper, _
    view As Overview, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The Overview

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions.Overview Method (IHtmlHelper, Overview, String, Object)

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Overview (
    HtmlHelper helper,
    Overview view,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As HtmlHelper, _
    view As Overview, _
```

```
    initFunction As String, _
    htmlAttributes As Object _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The Overview

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions.Overview Method (IHtmlHelper, String, String)

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Overview (
    HtmlHelper helper,
    string id,
    string diagramViewId
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Overview.

diagramViewId
The Id of the DiagramView

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions.Overview Method (IHtmlHelper, String, String, IDictionary(String, Object))

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Overview (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Overview

diagramViewId
The Id of the DiagramView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions.Overview Method (IHtmlHelper, String, String, Object)

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Overview (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Overview

diagramViewId
The Id of the DiagramView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions.Overview Method (IHtmlHelper, String, String, String)

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Overview (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    initFunction As String _
) As IHtmlContent
```

Parameters

helper

The HTML helper

id
The Id of the Overview

diagramViewId
The Id of the DiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewExtensions.Overview Method (IHtmlHelper, String, String, String, IDictionary(String, Object))

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Overview (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Overview

diagramViewId
The Id of the DiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members

OverviewExtensions.Overview Method (IHtmlHelper, String, String, String, Object)

Returns a Overview by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Overview (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Overview( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    initFunction As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Overview

diagramViewId
The Id of the DiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Overview Method Overload List
OverviewExtensions Members
OverviewExtensions Class
MindFusion.Diagramming.Mvc Namespace

OverviewTagHelper Class

Tag helper of Overview.

Syntax

C# Copy Code

```
public class OverviewTagHelper : TagHelper
```

Visual Basic Copy Code

```
Public Class OverviewTagHelper
    Inherits TagHelper
```

Remarks

This class implements the <mindfusion-overview> tag.

Inheritance Hierarchy

System.Object
  Microsoft.AspNetCore.Razor.TagHelpers.TagHelper
    MindFusion.Diagramming.Mvc.OverviewTagHelper

See Also

OverviewTagHelper Members
MindFusion.Diagramming.Mvc Namespace

OverviewTagHelper Members

Tag helper of Overview.

The following tables list the members exposed by the OverviewTagHelper type.

Public Constructors

	Name	Description
	OverviewTagHelper	

Public Properties

	Name	Description
	Component	
	Id	
	Style	

Public Methods

	Name	Description
	Process	

See Also

OverviewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

OverviewTagHelper Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public OverviewTagHelper ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

OverviewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

OverviewTagHelper Methods

OverviewTagHelper.Process Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override void Process (
    TagHelperContext context,
    TagHelperOutput output
)
```

Visual Basic Copy Code

```
Public Overrides Sub Process( _
    context As TagHelperContext, _
    output As TagHelperOutput _
)
```

Parameters

context

output

See Also

OverviewTagHelper Members
OverviewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

OverviewTagHelper Properties

OverviewTagHelper.Component Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Overview Component { get; set; }
```

Visual Basic Copy Code

```
Public Property Component As Overview
```

Property Value

See Also

OverviewTagHelper Members
OverviewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

OverviewTagHelper.Id Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

OverviewTagHelper Members
OverviewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

OverviewTagHelper.Style Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As String
```

Property Value

See Also

OverviewTagHelper Members
OverviewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

Palette Class

Represents a tool palette control that displays DiagramItem objects grouped into categories. Items from the palette can be added to a diagram by drag-and-drop.

Syntax

C# Copy Code

```
public class Palette
```

Visual Basic Copy Code

```
Public Class Palette
```

Remarks

Content Security Policy

To allow for stricter CSP rules, the library no longer loads CSS scripts automatically. If you use **Palette**, you must explicitly link the *common-ui.css* file from distribution's *themes* folder.

The **Palette** is an **Accordion** subclass that displays ItemListView components as its child panels. Each **ItemListView** displays the items of a palette category.

Call the AddCategory method to define a category and create its respective accordion pane. Call AddItem to add an item to specified category.

Drag-and-drop operations create a copy of the dragged item, and set its size to NewInstanceSize. The copy is created by calling item's clone method. You can override **clone** to copy property values of custom item subclasses.

Icons in the list are drawn by scaling **NewInstanceSize** down to IconSize. E.g. if a list icon is half the size of the diagram item it creates, its graphic elements such as cells, labels, strokes would be rendered at half scale too (as opposed to rendering with same label / stroke sizes but in smaller bounding rectangle). The scaling code assumes MeasureUnit of the target diagram and the palette have matching values.

Configuring the control

Palette can be created inside a Razor page either by invoking the Palette html helper, or by adding a <mindfusion-palette> tag helper:

Razor Copy Code

```
@addTagHelper *, MindFusion.Diagramming.Mvc
....
<mindfusion-palette
    component=(Palette)ViewBag.Palette
    style="width:100%; height:100%;">
</mindfusion-palette>
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.Palette

See Also

Palette Members
MindFusion.Diagramming.Mvc Namespace

Palette Members

Represents a tool palette control that displays DiagramItem objects grouped into categories. Items from the palette can be added to a diagram by drag-and-drop.

The following tables list the members exposed by the Palette type.

Public Constructors

	Name	Description
	Palette	Overloaded.

Public Properties

	Name	Description

	AddScriptReferences	Gets or sets a value indicating whether the control should render references to the client scripts.
	AutoClasses	Gets the registered client types with JSON templates.
	AutoConnectDistance	Gets or sets the maximum distance to a node when auto connecting links.
	AutoConnectLinks	Gets or sets whether links should be automatically connected to the nearest node.
	BackColor	Gets or sets the background color of the Palette control.
	CaptionFont	Gets or sets the font of the items captions.
	ClientScriptLocation	Gets or sets the URL to the MindFusion.Diagramming.js file.
	Columns	Gets or sets the number of columns in palette's layout.
	Enabled	Gets or sets a value indicating whether handling of mouse events is enabled.
	IconSize	Gets or sets the size of items displayed in the palette.
	Id	Gets or sets the id of the control.
	LabelPosition	Gets or sets a value indicating the position of labels relative to item icons.
	MeasureUnit	Gets or sets the unit of measure used to specify item sizes.
	MinVisibleFontSize	Gets or sets a threshold value that hides text if scaled font sizes become smaller.
	NewInstanceSize	Gets or sets the default size of items created by drag-and-drop from this palette.
	Orientation	Gets or sets the layout orientation of the control.
	Padding	Gets or sets the padding space between item's slot bounds and its contents.
	ShapeLibraryLocation	The location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path.
	TextColor	Gets or sets the color of the items captions.
	Theme	Gets or sets the theme used to style the control.

Public Methods

	Name	Description
	AddCategory	Overloaded.
	AddItem	Adds a DiagramItem to the displayed items collection.
	ControlLoadedScript	Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.
S	FromJson	Overloaded.
S	FromRequest	Initializes a new instance of the Palette class.
	Init	Renders the Palette's initializing scripts on a web page under the specified function name.
	LinkSelectedScript	Sets the name of a JavaScript function that should be invoked when a link is selected, either programmatically or by the user.
	LoadFromJson	Loads the specified JSON data.
	NodeSelectedScript	Sets the name of a JavaScript function that should be invoked when a node is selected, either programmatically or by the user.

	RegisterItemType	Overloaded.
	RegisterSession	Registers the session instance with the ItemTypeResolver used in custom item serialization.
	RemoveCategory	Removes a category from the palette.
	RemoveItem	Removes a DiagramItem from the displayed items collection.
	Render	Renders the Palette on a web page.
	SaveToJson	Serializes the Palette's data to JSON.
	SetAddScriptReferences	Sets a value indicating whether the control should render references to the client scripts.
	SetAutoConnectDistance	Sets the maximum distance to a node when auto connecting links.
	SetAutoConnectLinks	Sets whether links should be automatically connected to the nearest node.
	SetBackColor	Sets the background color of the Palette control.
	SetCaptionFont	Sets the font of the items captions.
	SetColumns	Sets the number of columns in control's layout.
	SetEnabled	Sets a value indicating whether handling of mouse events is enabled.
	SetIconSize	Sets the size of the items displayed in the view.
	SetLabelPosition	Sets a value indicating the position of labels relative to item icons.
	SetMinVisibleFontSize	Sets a threshold value that hides text if scaled font sizes become smaller.
	SetNewInstanceSize	Sets the default size with which items dragged onto a diagram will be created.
	SetOrientation	Sets the layout orientation of the control.
	SetTextColor	Sets the text color of the items captions.
	SetTheme	Sets the theme used to style the control.
	ShapeLibraryLoadedScript	Sets the name of a JavaScript function that should be invoked when a shape library is loaded.
	ToJson	Serializes the Palette's data to JSON.

See Also

Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette Constructor

Overload List

Name	Description
Palette ()	Initializes a new instance of the Palette class.
Palette (String)	Initializes a new instance of the Palette class.

See Also

Palette Constructor ()

Initializes a new instance of the Palette class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

Palette Constructor Overload List
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette Constructor (String)

Initializes a new instance of the Palette class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette (
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String _
)
```

Parameters

id
The Id of the Palette

See Also

Palette Constructor Overload List
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette Methods

Palette.AddCategory Method

Overload List

Name	Description
Palette.AddCategory (String)	Adds a new category to the palette.
Palette.AddCategory (String, String)	Adds a new category to the palette.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.AddCategory Method (String)

Adds a new category to the palette.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void AddCategory (
    string title
)
```

Visual Basic Copy Code

```
Public Sub AddCategory( _
    title As String _
)
```

Parameters

title
The title text of the category.

See Also

AddCategory Method Overload List
Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.AddCategory Method (String, String)

Adds a new category to the palette.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void AddCategory (
    string title,
    string imageUrl
)
```

Visual Basic Copy Code

```
Public Sub AddCategory( _
    title As String, _
    imageUrl As String _
)
```

Parameters

title
The title text of the category.

imageUrl
The URL of an image to be displayed in the category header.

See Also

AddCategory Method Overload List
Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.AddItem Method

Adds a DiagramItem to the displayed items collection.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void AddItem (
    DiagramItem item,
    string category,
    string caption
)
```

Visual Basic Copy Code

```
Public Sub AddItem( _
    item As DiagramItem, _
    category As String, _
    caption As String _
)
```

Parameters

item
The DiagramItem instance to add.

category
The category of the item.

caption
The caption to display for the item.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.ControlLoadedScript Method

Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette ControlLoadedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ControlLoadedScript( _
    value As String _
) As Palette
```

Parameters

value
The name of the function

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.FromJson Method

Overload List

Name	Description
Palette.FromJson (String)	Creates a Palette object from JSON.
Palette.FromJson (String, String)	Creates a Palette object from JSON.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.FromJson Method (String)

Creates a Palette object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static Palette FromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    json As String _
) As Palette
```

Parameters

json
The JSON string

Return Value

See Also

FromJson Method Overload List
Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.FromJson Method (String, String)

Creates a Palette object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static Palette FromJson (
    string id,
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    id As String, _
    json As String _
) As Palette
```

Parameters

id
The Id of the Palette

json
The JSON string

Return Value

See Also

FromJson Method Overload List
Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.FromRequest Method

Initializes a new instance of the Palette class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static Palette FromRequest (
    string id,
    HttpRequest request
)
```

Visual Basic Copy Code

```
Public Shared Function FromRequest( _
    id As String, _
    request As HttpRequest _
) As Palette
```

Parameters

id
The Id of the Palette

request
An HttpRequest object that contains the json data for the view.

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.Init Method

Renders the Palette's initializing scripts on a web page under the specified function name.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Init (
    string functionName
)
```

Visual Basic Copy Code

```
Public Function Init( _
    functionName As String _
) As String
```

Parameters

functionName

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.LinkSelectedScript Method

Sets the name of a JavaScript function that should be invoked when a link is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette LinkSelectedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function LinkSelectedScript( _
    value As String _
) As Palette
```

Parameters

value
The name of the function

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.LoadFromJson Method

Loads the specified JSON data.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void LoadFromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Sub LoadFromJson( _
    json As String _
)
```

Parameters

json
The JSON string

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.NodeSelectedScript Method

Sets the name of a JavaScript function that should be invoked when a node is selected, either programmatically or by the user.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette NodeSelectedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function NodeSelectedScript( _
    value As String _
) As Palette
```

Parameters

value
The name of the function

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.RegisterItemType Method

Overload List

Name	Description
Palette.RegisterItemType (Type, String, String, Int32)	Registers a custom item class for serialization support.
Palette.RegisterItemType (Type, String, String, Int32, String)	Registers a custom item class for serialization support.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.RegisterItemType Method (Type, String, String, Int32)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette RegisterItemType (
    Type itemClass,
    string classId,
    string clientClassId,
    int classVersion
)
```

Visual Basic Copy Code

```
Public Function RegisterItemType( _
    itemClass As Type, _
    classId As String, _
    clientClassId As String, _
    classVersion As Integer _
) As Palette
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when serializing, saving and loading items of the specified type.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

classVersion
A revision number of the item's class serialization format.

Return Value

See Also

RegisterItemType Method Overload List
Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.RegisterItemType Method (Type, String, String, Int32, String)

Registers a custom item class for serialization support.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette RegisterItemType (
    Type itemClass,
    string classId,
    string clientClassId,
    int classVersion,
    string jsonTemplate
)
```

Visual Basic Copy Code

```
Public Function RegisterItemType( _
    itemClass As Type, _
    classId As String, _
    clientClassId As String, _
    classVersion As Integer, _
    jsonTemplate As String _
) As Palette
```

Parameters

itemClass
A Type instance identifying the item class.

classId
A class identifier to use when serializing, saving and loading items of the specified type.

clientClassId
A class identifier to use when serializing items of the specified type on the client side.

classVersion
A revision number of the item's class serialization format.

jsonTemplate
A JSON string representation of the node contents.

Return Value

See Also

RegisterItemType Method Overload List
Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.RegisterSession Method

Registers the session instance with the ItemTypeResolver used in custom item serialization.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette RegisterSession (
    ISession session
)
```

Visual Basic Copy Code

```
Public Function RegisterSession( _
    session As ISession _
) As Palette
```

Parameters

session
An ISession instance.

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.RemoveCategory Method

Removes a category from the palette.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void RemoveCategory (
    string title
)
```

Visual Basic Copy Code

```
Public Sub RemoveCategory( _
    title As String _
)
```

Parameters

title
The title of the category to remove.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.RemoveItem Method

Removes a DiagramItem from the displayed items collection.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void RemoveItem (
    DiagramItem item
)
```

Visual Basic Copy Code

```
Public Sub RemoveItem( _
    item As DiagramItem _
)
```

Parameters

item
The DiagramItem instance to remove.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.Render Method

Renders the Palette on a web page.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Render ()
```

Visual Basic Copy Code

```
Public Function Render () As String
```

Return Value

See Also

Palette.SaveToJson Method

Serializes the Palette's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string SaveToJson ()
```

Visual Basic Copy Code

```
Public Function SaveToJson () As String
```

Return Value

The JSON string.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetAddScriptReferences Method

Sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetAddScriptReferences (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAddScriptReferences( _
    value As Boolean _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetAutoConnectDistance Method

Sets the maximum distance to a node when auto connecting links.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetAutoConnectDistance (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetAutoConnectDistance( _
    value As Single _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetAutoConnectLinks Method

Sets whether links should be automatically connected to the nearest node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetAutoConnectLinks (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAutoConnectLinks( _
    value As Boolean _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetBackColor Method

Sets the background color of the Palette control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetBackColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetBackColor( _
    value As Color _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetCaptionFont Method

Sets the font of the items captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetCaptionFont (
    Font value
)
```

Visual Basic Copy Code

```
Public Function SetCaptionFont( _
    value As Font _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetColumns Method

Sets the number of columns in control's layout.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetColumns (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetColumns( _
    value As Integer _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetEnabled Method

Sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetEnabled (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetEnabled( _
    value As Boolean _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetIconSize Method

Sets the size of the items displayed in the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetIconSize (
    SizeF value
)
```

Visual Basic Copy Code

```
Public Function SetIconSize( _
    value As SizeF _
) As Palette
```

Parameters

value
The size of the items

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetLabelPosition Method

Sets a value indicating the position of labels relative to item icons.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetLabelPosition (
    LabelPosition value
)
```

Visual Basic Copy Code

```
Public Function SetLabelPosition( _
    value As LabelPosition _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetMinVisibleFontSize Method

Sets a threshold value that hides text if scaled font sizes become smaller.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetMinVisibleFontSize (
    float value
)
```

Visual Basic Copy Code

```
Public Function SetMinVisibleFontSize( _
    value As Single _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetNewInstanceSize Method

Sets the default size with which items dragged onto a diagram will be created.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetNewInstanceSize (
    SizeF value
)
```

Visual Basic Copy Code

```
Public Function SetNewInstanceSize( _
    value As SizeF _
) As Palette
```

Parameters

value
The size of the items

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetOrientation Method

Sets the layout orientation of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetOrientation (
    Orientation value
)
```

Visual Basic Copy Code

```
Public Function SetOrientation( _
    value As Orientation _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetTextColor Method

Sets the text color of the items captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetTextColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetTextColor( _
    value As Color _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.SetTheme Method

Sets the theme used to style the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette SetTheme (
    string value
)
```

Visual Basic Copy Code

```
Public Function SetTheme( _
    value As String _
) As Palette
```

Parameters

value

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.ShapeLibraryLoadedScript Method

Sets the name of a JavaScript function that should be invoked when a shape library is loaded.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette ShapeLibraryLoadedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ShapeLibraryLoadedScript( _
    value As String _
) As Palette
```

Parameters

value
The name of the function

Return Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.ToJson Method

Serializes the Palette's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static string ToJson (
    Palette view
)
```

Visual Basic Copy Code

```
Public Shared Function ToJson( _
    view As Palette _
) As String
```

Parameters

view

Return Value

The JSON string.

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette Properties

Palette.AddScriptReferences Property

Gets or sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AddScriptReferences { get; set; }
```

Visual Basic Copy Code

```
Public Property AddScriptReferences As Boolean
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.AutoClasses Property

Gets the registered client types with JSON templates.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Dictionary<string, string> AutoClasses { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property AutoClasses As Dictionary(Of String, String)
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.AutoConnectDistance Property

Gets or sets the maximum distance to a node when auto connecting links.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float AutoConnectDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoConnectDistance As Single
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.AutoConnectLinks Property

Gets or sets whether links should be automatically connected to the nearest node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AutoConnectLinks { get; set; }
```

Visual Basic Copy Code

```
Public Property AutoConnectLinks As Boolean
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.BackColor Property

Gets or sets the background color of the Palette control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color BackColor { get; set; }
```

Visual Basic Copy Code

```
Public Property BackColor As Color
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.CaptionFont Property

Gets or sets the font of the items captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Font CaptionFont { get; set; }
```

Visual Basic Copy Code

```
Public Property CaptionFont As Font
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.ClientScriptLocation Property

Gets or sets the URL to the MindFusion.Diagramming.js file.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string ClientScriptLocation { get; set; }
```

Visual Basic Copy Code

```
Public Property ClientScriptLocation As String
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.Columns Property

Gets or sets the number of columns in palette's layout.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Columns { get; set; }
```

Visual Basic Copy Code

```
Public Property Columns As Integer
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.Enabled Property

Gets or sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool Enabled { get; set; }
```

Visual Basic Copy Code

```
Public Property Enabled As Boolean
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.IconSize Property

Gets or sets the size of items displayed in the palette.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public SizeF IconSize { get; set; }
```

Visual Basic Copy Code

```
Public Property IconSize As SizeF
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.Id Property

Gets or sets the id of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.LabelPosition Property

Gets or sets a value indicating the position of labels relative to item icons.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public LabelPosition LabelPosition { get; set; }
```

Visual Basic Copy Code

```
Public Property LabelPosition As LabelPosition
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.MeasureUnit Property

Gets or sets the unit of measure used to specify item sizes.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public MeasureUnit MeasureUnit { get; set; }
```

Visual Basic Copy Code

```
Public Property MeasureUnit As MeasureUnit
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.MinVisibleFontSize Property

Gets or sets a threshold value that hides text if scaled font sizes become smaller.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public float MinVisibleFontSize { get; set; }
```

Visual Basic Copy Code

```
Public Property MinVisibleFontSize As Single
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.NewInstanceSize Property

Gets or sets the default size of items created by drag-and-drop from this palette.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public SizeF NewInstanceSize { get; set; }
```

Visual Basic Copy Code

```
Public Property NewInstanceSize As SizeF
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.Orientation Property

Gets or sets the layout orientation of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Orientation Orientation { get; set; }
```

Visual Basic Copy Code

```
Public Property Orientation As Orientation
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.Padding Property

Gets or sets the padding space between item's slot bounds and its contents.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Padding { get; set; }
```

Visual Basic Copy Code

```
Public Property Padding As Integer
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.ShapeLibraryLocation Property

The location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string ShapeLibraryLocation { get; set; }
```

Visual Basic Copy Code

```
Public Property ShapeLibraryLocation As String
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

Palette.TextColor Property

Gets or sets the color of the items captions.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color TextColor { get; set; }
```

Visual Basic Copy Code

```
Public Property TextColor As Color
```

Property Value

See Also

Palette.Theme Property

Gets or sets the theme used to style the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Theme { get; set; }
```

Visual Basic Copy Code

```
Public Property Theme As String
```

Property Value

See Also

Palette Members
Palette Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions Class

Contains extension methods of Palette.

Syntax

C# Copy Code

```
public sealed class PaletteExtensions
```

Visual Basic Copy Code

```
Public NotInheritable Class PaletteExtensions
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.PaletteExtensions

See Also

PaletteExtensions Members
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions Members

Contains extension methods of Palette.

The following tables list the members exposed by the PaletteExtensions type.

Public Methods

	Name	Description

	Generate	
	Palette	Overloaded.

See Also

PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions Methods

PaletteExtensions.Generate Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Generate (
    Palette view
)
```

Visual Basic Copy Code

```
Public Shared Function Generate( _
    view As Palette _
) As IHtmlContent
```

Parameters

view

Return Value

See Also

PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method

Overload List

Name	Description
PaletteExtensions.Palette (IHtmlHelper, Palette)	Returns a Palette by using the specified HTML helper.
PaletteExtensions.Palette (IHtmlHelper, Palette, IDictionary(String, Object))	Returns a Palette by using the specified HTML helper.
PaletteExtensions.Palette (IHtmlHelper, Palette, Object)	Returns a Palette by using the specified HTML helper.
PaletteExtensions.Palette (IHtmlHelper, Palette, String)	Returns a Palette by using the specified HTML helper.
PaletteExtensions.Palette (IHtmlHelper, Palette, String, IDictionary(String, Object))	Returns a Palette by using the specified HTML helper.

PaletteExtensions.Palette (IHtmlHelper, Palette, String, Object)	Returns a Palette by using the specified HTML helper.
PaletteExtensions.Palette (IHtmlHelper, String)	Returns a Palette by using the specified HTML helper.
PaletteExtensions.Palette (IHtmlHelper, String, IDictionary(String, Object))	Returns a Palette by using the specified HTML helper.
PaletteExtensions.Palette (IHtmlHelper, String, Object)	Returns a Palette by using the specified HTML helper.
PaletteExtensions.Palette (IHtmlHelper, String, String)	Returns a Palette by using the specified HTML helper.
PaletteExtensions.Palette (IHtmlHelper, String, String, IDictionary(String, Object))	Returns a Palette by using the specified HTML helper.
PaletteExtensions.Palette (IHtmlHelper, String, String, Object)	Returns a Palette by using the specified HTML helper.

See Also

PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, Palette)

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    Palette view
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    view As Palette _
) As IHtmlContent
```

Parameters

helper
The HTML helper

view
The Palette

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, Palette, IDictionary(String, Object))

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    Palette view,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    view As Palette, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

view
The Palette

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, Palette, Object)

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    Palette view,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    view As Palette, _
```

```
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

view
The Palette

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, Palette, String)

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    Palette view,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    view As Palette, _
    initFunction As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

view
The Palette

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, Palette, String, IDictionary(String, Object))

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    Palette view,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    view As Palette, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

view
The Palette

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, Palette, String, Object)

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    Palette view,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    view As Palette, _
    initFunction As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

view
The Palette

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, String)

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    string id
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    id As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Palette

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, String, IDictionary(String, Object))

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    string id,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    id As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Palette

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, String, Object)

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    string id,
```

```
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    id As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Palette

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, String, String)

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    string id,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    id As String, _
    initFunction As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Palette

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, String, String, IDictionary(String, Object))

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    string id,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    id As String, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Palette

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteExtensions.Palette Method (IHtmlHelper, String, String, Object)

Returns a Palette by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Palette (
    IHtmlHelper helper,
    string id,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Palette( _
    helper As IHtmlHelper, _
    id As String, _
    initFunction As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Palette

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Palette Method Overload List
PaletteExtensions Members
PaletteExtensions Class
MindFusion.Diagramming.Mvc Namespace

PaletteTagHelper Class

Tag helper of Palette.

Syntax

C# Copy Code

```
public class PaletteTagHelper : TagHelper
```

Visual Basic Copy Code

```
Public Class PaletteTagHelper
    Inherits TagHelper
```

Remarks

This class implements the <mindfusion-palette> tag.

Inheritance Hierarchy

System.Object
  Microsoft.AspNetCore.Razor.TagHelpers.TagHelper
    MindFusion.Diagramming.Mvc.PaletteTagHelper

See Also

PaletteTagHelper Members
MindFusion.Diagramming.Mvc Namespace

PaletteTagHelper Members

Tag helper of Palette.

The following tables list the members exposed by the PaletteTagHelper type.

Public Constructors

	Name	Description
	PaletteTagHelper	

Public Properties

	Name	Description
	Component	
	Id	
	Style	

Public Methods

	Name	Description
	Process	

See Also

PaletteTagHelper Class
MindFusion.Diagramming.Mvc Namespace

PaletteTagHelper Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public PaletteTagHelper ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

PaletteTagHelper Class
MindFusion.Diagramming.Mvc Namespace

PaletteTagHelper Methods

PaletteTagHelper.Process Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override void Process (
    TagHelperContext context,
    TagHelperOutput output
)
```

Visual Basic Copy Code

```
Public Overrides Sub Process( _
    context As TagHelperContext, _
    output As TagHelperOutput _
)
```

Parameters

context

output

See Also

PaletteTagHelper Members
PaletteTagHelper Class
MindFusion.Diagramming.Mvc Namespace

PaletteTagHelper Properties

PaletteTagHelper.Component Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Palette Component { get; set; }
```

Visual Basic Copy Code

```
Public Property Component As Palette
```

Property Value

See Also

PaletteTagHelper Members
PaletteTagHelper Class
MindFusion.Diagramming.Mvc Namespace

PaletteTagHelper.Id Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

PaletteTagHelper Members
PaletteTagHelper Class
MindFusion.Diagramming.Mvc Namespace

PaletteTagHelper.Style Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As String
```

Property Value

See Also

PaletteTagHelper Members
PaletteTagHelper Class
MindFusion.Diagramming.Mvc Namespace

Ruler Class

Provides horizontal and vertical scales that helps users measure diagram items.

Syntax

C# Copy Code

```
public class Ruler
```

Visual Basic Copy Code

```
Public Class Ruler
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.Ruler

See Also

Ruler Members
MindFusion.Diagramming.Mvc Namespace

Ruler Members

Provides horizontal and vertical scales that helps users measure diagram items.

The following tables list the members exposed by the Ruler type.

Public Constructors

	Name	Description
	Ruler	Overloaded.

Public Properties

	Name	Description
	AddScriptReferences	Gets or sets a value indicating whether the control should render references to the client scripts.
	BackColor	Gets or sets the background color of the Ruler control.
	ClientScriptLocation	Gets or sets the URL to the MindFusion.Diagramming.js file.
	DiagramView	Gets or sets the id of the DiagramView.
	EnableGuides	Gets or sets a value indicating whether alignment guides are active.
	ForeColor	Gets or sets the foreground color of the control.
	GuideColor	Gets or sets the color used to paint guides.
	HorizontalScaleVisible	Gets or sets a value indicating whether the horizontal ruler scale is visible.
	Id	Gets or sets the id of the view.
	NegatedX	Gets or sets whether the horizontal scale should show negated values, simulating that the coordinate system's X axis grows to the left.
	NegatedY	Gets or sets whether the vertical scale should show negated values, simulating that the coordinate system's Y axis grows to the top.
	PointerColor	Gets or sets the color used to paint cursor projections.
	ProjectionColor	Gets or sets the color used to paint the projection of the active node.
	ProjectRotatedBounds	Gets or sets whether the projection of nodes on ruler's scales rotates with them.
	TextColor	Gets or sets the color used to paint texts in the ruler.
	Unit	Gets or sets the measure unit used by the ruler.
	VerticalScaleVisible	Gets or sets a value indicating whether the vertical ruler scale is visible.

Public Methods

	Name	Description
	ControlLoadedScript	Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.
S	FromJson	Overloaded.
S	FromRequest	Overloaded.
	Init	Renders the Ruler's initializing scripts on a web page under the specified function name.
	LoadFromJson	Loads the specified JSON data.
	Render	Renders the **Ruler** on a web page.

	SaveToJson	Serializes the Ruler's data to JSON.
	SetAddScriptReferences	Sets a value indicating whether the control should render references to the client scripts.
	SetBackColor	Sets the background color of the **Ruler** control.
	SetDiagramView	Sets the id of the **DiagramView**.
	SetEnableGuides	Sets a value indicating whether alignment guides are active.
	SetForeColor	Sets the foreground color of the control.
	SetGuideColor	Sets the color used to paint guides.
	SetHorizontalScaleVisible	Sets a value indicating whether the horizontal ruler scale is visible.
	SetNegatedX	Sets whether the horizontal scale should show negated values, simulating that the coordinate system's X axis grows to the left.
	SetNegatedY	Sets whether the vertical scale should show negated values, simulating that the coordinate system's Y axis grows to the top.
	SetPointerColor	Sets the color used to paint cursor projections.
	SetProjectionColor	Sets the color used to paint the projection of the active node.
	SetProjectRotatedBounds	Sets whether the projection of nodes on ruler's scales rotates with them.
	SetTextColor	Sets the color used to paint texts in the ruler.
	SetUnit	Sets the measure unit used by the ruler.
	SetVerticalScaleVisible	Sets a value indicating whether the vertical ruler scale is visible.
	ToJson	Serializes the Ruler's data to JSON.

See Also

Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler Constructor

Overload List

Name	Description
Ruler ()	Creates an instance of Ruler.
Ruler (String)	Creates an instance of **Ruler**.
Ruler (String, String)	Creates an instance of **Ruler** for a DiagramView.

See Also

Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler Constructor ()

Creates an instance of Ruler.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

Ruler Constructor Overload List
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler Constructor (String)

Creates an instance of Ruler.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler (
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String _
)
```

Parameters

id

See Also

Ruler Constructor Overload List
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler Constructor (String, String)

Creates an instance of Ruler for a DiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler (
    string id,
    string diagramViewId
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String, _
    diagramViewId As String _
)
```

Parameters

id
The Id of the Ruler

diagramViewId
The Id of the DiagramView

See Also

Ruler Constructor Overload List
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler Methods

Ruler.ControlLoadedScript Method

Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler ControlLoadedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function ControlLoadedScript( _
    value As String _
) As Ruler
```

Parameters

value
The name of the function

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.FromJson Method

Overload List

Name	Description
Ruler.FromJson (String)	Creates a Ruler object from JSON.
Ruler.FromJson (String, String)	Creates a Ruler object from JSON.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.FromJson Method (String)

Creates a Ruler object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static Ruler FromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    json As String _
) As Ruler
```

Parameters

json
The JSON string

Return Value

See Also

FromJson Method Overload List
Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.FromJson Method (String, String)

Creates a Ruler object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static Ruler FromJson (
    string id,
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    id As String, _
    json As String _
) As Ruler
```

Parameters

id

The Id of the Ruler

json
The JSON string

Return Value

See Also

FromJson Method Overload List
Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.FromRequest Method

Overload List

Name	Description
Ruler.FromRequest (String, HttpRequest)	Creates an instance of Ruler.
Ruler.FromRequest (String, HttpRequestBase)	Creates an instance of Ruler.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.FromRequest Method (String, HttpRequest)

Creates an instance of Ruler.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static Ruler FromRequest (
    string id,
    HttpRequest request
)
```

Visual Basic Copy Code

```
Public Shared Function FromRequest( _
    id As String, _
    request As HttpRequest _
) As Ruler
```

Parameters

id
The Id of the Ruler

request
An HttpRequest object that contains the json data for the view.

Return Value

See Also

FromRequest Method Overload List
Ruler Members

Ruler.FromRequest Method (String, HttpRequestBase)

Creates an instance of Ruler.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static Ruler FromRequest (
    string id,
    HttpRequestBase request
)
```

Visual Basic Copy Code

```
Public Shared Function FromRequest( _
    id As String, _
    request As HttpRequestBase _
) As Ruler
```

Parameters

id
The Id of the Ruler

request
An HttpRequest object that contains the json data for the view.

Return Value

See Also

FromRequest Method Overload List
Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.Init Method

Renders the Ruler's initializing scripts on a web page under the specified function name.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Init (
    string functionName
)
```

Visual Basic Copy Code

```
Public Function Init( _
    functionName As String _
) As String
```

Parameters

functionName

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.LoadFromJson Method

Loads the specified JSON data.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void LoadFromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Sub LoadFromJson( _
    json As String _
)
```

Parameters

json
The JSON string

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.Render Method

Renders the Ruler on a web page.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Render ()
```

Visual Basic Copy Code

```
Public Function Render () As String
```

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SaveToJson Method

Serializes the Ruler's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string SaveToJson ()
```

Visual Basic Copy Code

```
Public Function SaveToJson () As String
```

Return Value

The JSON string.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetAddScriptReferences Method

Sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetAddScriptReferences (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAddScriptReferences( _
    value As Boolean _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetBackColor Method

Sets the background color of the Ruler control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetBackColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetBackColor( _
    value As Color _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetDiagramView Method

Sets the id of the DiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetDiagramView (
    string value
)
```

Visual Basic Copy Code

```
Public Function SetDiagramView( _
    value As String _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetEnableGuides Method

Sets a value indicating whether alignment guides are active.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetEnableGuides (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetEnableGuides( _
    value As Boolean _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetForeColor Method

Sets the foreground color of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetForeColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetForeColor( _
    value As Color _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetGuideColor Method

Sets the color used to paint guides.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetGuideColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetGuideColor( _
    value As Color _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetHorizontalScaleVisible Method

Sets a value indicating whether the horizontal ruler scale is visible.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetHorizontalScaleVisible (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetHorizontalScaleVisible( _
    value As Boolean _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetNegatedX Method

Sets whether the horizontal scale should show negated values, simulating that the coordinate system's X axis grows to the left.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetNegatedX (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetNegatedX( _
    value As Boolean _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetNegatedY Method

Sets whether the vertical scale should show negated values, simulating that the coordinate system's Y axis grows to the top.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetNegatedY (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetNegatedY( _
    value As Boolean _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetPointerColor Method

Sets the color used to paint cursor projections.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetPointerColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetPointerColor( _
    value As Color _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetProjectionColor Method

Sets the color used to paint the projection of the active node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetProjectionColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetProjectionColor( _
    value As Color _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetProjectRotatedBounds Method

Sets whether the projection of nodes on ruler's scales rotates with them.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetProjectRotatedBounds (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetProjectRotatedBounds( _
    value As Boolean _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetTextColor Method

Sets the color used to paint texts in the ruler.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetTextColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetTextColor( _
    value As Color _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetUnit Method

Sets the measure unit used by the ruler.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetUnit (
    MeasureUnit value
)
```

Visual Basic Copy Code

```
Public Function SetUnit( _
    value As MeasureUnit _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.SetVerticalScaleVisible Method

Sets a value indicating whether the vertical ruler scale is visible.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler SetVerticalScaleVisible (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetVerticalScaleVisible( _
    value As Boolean _
) As Ruler
```

Parameters

value

Return Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.ToJson Method

Serializes the Ruler's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static string ToJson (
    Ruler ruler
)
```

Visual Basic Copy Code

```
Public Shared Function ToJson( _
    ruler As Ruler _
) As String
```

Parameters

ruler

Return Value

The JSON string.

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler Properties

Ruler.AddScriptReferences Property

Gets or sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AddScriptReferences { get; set; }
```

Visual Basic Copy Code

```
Public Property AddScriptReferences As Boolean
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.BackColor Property

Gets or sets the background color of the Ruler control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public Color BackColor { get; set; }
```

Visual Basic

```
Public Property BackColor As Color
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.ClientScriptLocation Property

Gets or sets the URL to the MindFusion.Diagramming.js file.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public string ClientScriptLocation { get; set; }
```

Visual Basic

```
Public Property ClientScriptLocation As String
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.DiagramView Property

Gets or sets the id of the DiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public string DiagramView { get; set; }
```

Visual Basic

```
Public Property DiagramView As String
```

Property Value

See Also

Ruler Members
Ruler Class

Ruler.EnableGuides Property

Gets or sets a value indicating whether alignment guides are active.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool EnableGuides { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableGuides As Boolean
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.ForeColor Property

Gets or sets the foreground color of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color ForeColor { get; set; }
```

Visual Basic Copy Code

```
Public Property ForeColor As Color
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.GuideColor Property

Gets or sets the color used to paint guides.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color GuideColor { get; set; }
```

Visual Basic Copy Code

```
Public Property GuideColor As Color
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.HorizontalScaleVisible Property

Gets or sets a value indicating whether the horizontal ruler scale is visible.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool HorizontalScaleVisible { get; set; }
```

Visual Basic Copy Code

```
Public Property HorizontalScaleVisible As Boolean
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.Id Property

Gets or sets the id of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.NegatedX Property

Gets or sets whether the horizontal scale should show negated values, simulating that the coordinate system's X axis grows to the left.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool NegatedX { get; set; }
```

Visual Basic Copy Code

```
Public Property NegatedX As Boolean
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.NegatedY Property

Gets or sets whether the vertical scale should show negated values, simulating that the coordinate system's Y axis grows to the top.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool NegatedY { get; set; }
```

Visual Basic Copy Code

```
Public Property NegatedY As Boolean
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.PointerColor Property

Gets or sets the color used to paint cursor projections.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color PointerColor { get; set; }
```

Visual Basic Copy Code

```
Public Property PointerColor As Color
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.ProjectionColor Property

Gets or sets the color used to paint the projection of the active node.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color ProjectionColor { get; set; }
```

Visual Basic Copy Code

```
Public Property ProjectionColor As Color
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.ProjectRotatedBounds Property

Gets or sets whether the projection of nodes on ruler's scales rotates with them.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool ProjectRotatedBounds { get; set; }
```

Visual Basic Copy Code

```
Public Property ProjectRotatedBounds As Boolean
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.TextColor Property

Gets or sets the color used to paint texts in the ruler.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color TextColor { get; set; }
```

Visual Basic Copy Code

```
Public Property TextColor As Color
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.Unit Property

Gets or sets the measure unit used by the ruler.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public MeasureUnit Unit { get; set; }
```

Visual Basic Copy Code

```
Public Property Unit As MeasureUnit
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

Ruler.VerticalScaleVisible Property

Gets or sets a value indicating whether the vertical ruler scale is visible.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool VerticalScaleVisible { get; set; }
```

Visual Basic Copy Code

```
Public Property VerticalScaleVisible As Boolean
```

Property Value

See Also

Ruler Members
Ruler Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions Class

Contains extension methods of Ruler.

Syntax

C# Copy Code

```
public sealed class RulerExtensions
```

Visual Basic Copy Code

```
Public NotInheritable Class RulerExtensions
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.RulerExtensions

See Also

RulerExtensions Members
MindFusion.Diagramming.Mvc Namespace

RulerExtensions Members

Contains extension methods of Ruler.

The following tables list the members exposed by the RulerExtensions type.

Public Methods

	Name	Description
	Ruler	Overloaded.

See Also

RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions Methods

RulerExtensions.Ruler Method

Overload List

Name	Description

RulerExtensions.Ruler (IHtmlHelper, Ruler)	Returns a Ruler by using the specified HTML helper.
RulerExtensions.Ruler (IHtmlHelper, Ruler, IDictionary(String, Object))	Returns a **Ruler** by using the specified HTML helper.
RulerExtensions.Ruler (IHtmlHelper, Ruler, Object)	Returns a **Ruler** by using the specified HTML helper.
RulerExtensions.Ruler (IHtmlHelper, Ruler, String)	Returns a **Ruler** by using the specified HTML helper.
RulerExtensions.Ruler (IHtmlHelper, Ruler, String, IDictionary(String, Object))	Returns a **Ruler** by using the specified HTML helper.
RulerExtensions.Ruler (IHtmlHelper, Ruler, String, Object)	Returns a **Ruler** by using the specified HTML helper.
RulerExtensions.Ruler (IHtmlHelper, String, String)	Returns a **Ruler** by using the specified HTML helper.
RulerExtensions.Ruler (IHtmlHelper, String, String, IDictionary(String, Object))	Returns a **Ruler** by using the specified HTML helper.
RulerExtensions.Ruler (IHtmlHelper, String, String, Object)	Returns a **Ruler** by using the specified HTML helper.
RulerExtensions.Ruler (IHtmlHelper, String, String, String)	Returns a **Ruler** by using the specified HTML helper.
RulerExtensions.Ruler (IHtmlHelper, String, String, String, IDictionary(String, Object))	Returns a **Ruler** by using the specified HTML helper.
RulerExtensions.Ruler (IHtmlHelper, String, String, String, Object)	Returns a **Ruler** by using the specified HTML helper.

See Also

RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, Ruler)

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Ruler (
    HtmlHelper helper,
    Ruler view
)
```

Visual Basic Copy Code

```
Public Shared Function Ruler( _
    helper As HtmlHelper, _
    view As Ruler _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The Ruler

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, Ruler, IDictionary(String, Object))

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Ruler (
    HtmlHelper helper,
    Ruler view,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Ruler( _
    helper As HtmlHelper, _
    view As Ruler, _
    htmlAttributes As IDictionary(Of String, Object) _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The Ruler

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, Ruler, Object)

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Ruler (
    HtmlHelper helper,
    Ruler view,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Ruler( _
    helper As HtmlHelper, _
    view As Ruler, _
    htmlAttributes As Object _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The Ruler

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, Ruler, String)

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Ruler (
    HtmlHelper helper,
    Ruler view,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function Ruler( _
    helper As HtmlHelper, _
    view As Ruler, _
    initFunction As String _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The Ruler

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, Ruler, String, IDictionary(String, Object))

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString Ruler (
    HtmlHelper helper,
    Ruler view,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Ruler( _
    helper As HtmlHelper, _
    view As Ruler, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The Ruler

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, Ruler, String, Object)

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public static MvcHtmlString Ruler (
    HtmlHelper helper,
    Ruler view,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic

```
Public Shared Function Ruler( _
    helper As HtmlHelper, _
    view As Ruler, _
    initFunction As String, _
    htmlAttributes As Object _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

view
The Ruler

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, String, String)

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public static IHtmlContent Ruler (
    IHtmlHelper helper,
    string id,
    string diagramViewId
)
```

Visual Basic

```
Public Shared Function Ruler( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Ruler

diagramViewId
The Id of the DiagramView

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, String, String, IDictionary(String, Object))

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Ruler (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Ruler( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Ruler

diagramViewId
The Id of the DiagramView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, String, String, Object)

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Ruler (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Ruler( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Ruler

diagramViewId
The Id of the DiagramView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, String, String, String)

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Ruler (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function Ruler( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
```

```
    initFunction As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Ruler

diagramViewId
The Id of the DiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, String, String, String, IDictionary(String, Object))

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Ruler (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Ruler( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Ruler

diagramViewId
The Id of the DiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerExtensions.Ruler Method (IHtmlHelper, String, String, String, Object)

Returns a Ruler by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent Ruler (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function Ruler( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    initFunction As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the Ruler

diagramViewId
The Id of the DiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

Ruler Method Overload List
RulerExtensions Members
RulerExtensions Class
MindFusion.Diagramming.Mvc Namespace

RulerTagHelper Class

Tag helper of Ruler.

Syntax

C# Copy Code

```
public class RulerTagHelper : TagHelper
```

Visual Basic Copy Code

```
Public Class RulerTagHelper
    Inherits TagHelper
```

Remarks

This class implements the <mindfusion-ruler> tag.

Inheritance Hierarchy

System.Object
  Microsoft.AspNetCore.Razor.TagHelpers.TagHelper
    MindFusion.Diagramming.Mvc.RulerTagHelper

See Also

RulerTagHelper Members
MindFusion.Diagramming.Mvc Namespace

RulerTagHelper Members

Tag helper of Ruler.

The following tables list the members exposed by the RulerTagHelper type.

Public Constructors

	Name	Description
	RulerTagHelper	

Public Properties

	Name	Description
	Component	
	Id	
	Style	

Public Methods

	Name	Description
	Process	

See Also

RulerTagHelper Class
MindFusion.Diagramming.Mvc Namespace

RulerTagHelper Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public RulerTagHelper ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

RulerTagHelper Class
MindFusion.Diagramming.Mvc Namespace

RulerTagHelper Methods

RulerTagHelper.Process Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override void Process (
    TagHelperContext context,
    TagHelperOutput output
)
```

Visual Basic Copy Code

```
Public Overrides Sub Process( _
    context As TagHelperContext, _
    output As TagHelperOutput _
)
```

Parameters

context

output

See Also

RulerTagHelper Members
RulerTagHelper Class
MindFusion.Diagramming.Mvc Namespace

RulerTagHelper Properties

RulerTagHelper.Component Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Ruler Component { get; set; }
```

Visual Basic Copy Code

```
Public Property Component As Ruler
```

Property Value

See Also

RulerTagHelper Members
RulerTagHelper Class
MindFusion.Diagramming.Mvc Namespace

RulerTagHelper.Id Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

RulerTagHelper Members
RulerTagHelper Class
MindFusion.Diagramming.Mvc Namespace

RulerTagHelper.Style Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As String
```

Property Value

See Also

RulerTagHelper Members
RulerTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ScaleMode Enumeration

Defines Overview scaling modes.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public enum ScaleMode
```

Visual Basic Copy Code

```
Public Enum ScaleMode
```

Members

	Member name	Description
	FitAll	Scale the Overview dynamically to fit the entire diagram.
	CombinedScales	Combine diagram's ZoomFactor with overview's ScaleFactor.
	FixedScale	Use fixed scale specified via ScaleFactor, ignoring diagram's zoom level.

See Also

MindFusion.Diagramming.Mvc Namespace

ShapeNodeConverter Class

Serializes ShapeNode instances.

Syntax

C# Copy Code

```
public class ShapeNodeConverter : DiagramNodeConverter
```

Visual Basic Copy Code

```
Public Class ShapeNodeConverter
    Inherits DiagramNodeConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.DiagramItemConverter
      MindFusion.Diagramming.Mvc.DiagramNodeConverter
        MindFusion.Diagramming.Mvc.ShapeNodeConverter

See Also

ShapeNodeConverter Members
MindFusion.Diagramming.Mvc Namespace

ShapeNodeConverter Members

Serializes ShapeNode instances.

The following tables list the members exposed by the ShapeNodeConverter type.

Public Constructors

	Name	Description
	ShapeNodeConverter	Creates a new instance of the ShapeNodeConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

ShapeNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ShapeNodeConverter Constructor

Creates a new instance of the ShapeNodeConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ShapeNodeConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

ShapeNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ShapeNodeConverter Methods

ShapeNodeConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

ShapeNodeConverter Members
ShapeNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ShapeNodeConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

ShapeNodeConverter Members
ShapeNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ShapeNodeConverter Properties

ShapeNodeConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

ShapeNodeConverter Members
ShapeNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

SvgNodeConverter Class

Serializes SvgNode instances.

Syntax

C# Copy Code

```
public class SvgNodeConverter : DiagramNodeConverter
```

Visual Basic Copy Code

```
Public Class SvgNodeConverter
    Inherits DiagramNodeConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.DiagramItemConverter
      MindFusion.Diagramming.Mvc.DiagramNodeConverter
        MindFusion.Diagramming.Mvc.SvgNodeConverter

See Also

SvgNodeConverter Members
MindFusion.Diagramming.Mvc Namespace

SvgNodeConverter Members

Serializes SvgNode instances.

The following tables list the members exposed by the SvgNodeConverter type.

Public Constructors

	Name	Description
	SvgNodeConverter	Creates a new instance of the SvgNodeConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

SvgNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

SvgNodeConverter Constructor

Creates a new instance of the SvgNodeConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public SvgNodeConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

SvgNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

SvgNodeConverter Methods

SvgNodeConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

SvgNodeConverter Members
SvgNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

SvgNodeConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C#

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

SvgNodeConverter Members
SvgNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

SvgNodeConverter Properties

SvgNodeConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

SvgNodeConverter Members
SvgNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

TabAlignment Enumeration

Specifies the alignment of a tab strip relative to its container control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public enum TabAlignment
```

Visual Basic Copy Code

```
Public Enum TabAlignment
```

Members

	Member name	Description
	None	The tab strip is not shown.
	Top	The tab strip is located at the top of the control.
	Bottom	The tab strip is located at the bottom of the control.

See Also

MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView Class

Represents a DiagramView that can display DiagramDocument objects. The individual diagram pages can be activated through the tabs in the tray located at one of the view sides.

Syntax

C# Copy Code

```
public class TabbedDiagramView : DiagramView
```

Visual Basic Copy Code

```
Public Class TabbedDiagramView
    Inherits DiagramView
```

Remarks

Use the TabbedDiagramView html helper to add this control to a page:

Index.cshtml Copy Code

```
@Html.TabbedDiagramView(
    (TabbedDiagramView)ViewBag.DiagramView,
    new { style = "width:100%;height:100%;" })
```

Setting up the view from code-behind looks like this:

Controller.cs Copy Code

```
public IActionResult Index()
{
    TabbedDiagramView view = TabbedDiagramView.FromRequest("diagramView", Request);
    ViewBag.DiagramView = view;

    if (!Request.HasFormContentType)
    {
        view.TabAlignment = TabAlignment.Top;

        var doc = view.Document;

        var p1 = new DiagramPage();
        p1.Title = "diagram1";
        p1.TitleColor = Color.Red;
        var node = p1.Factory.CreateShapeNode(10, 10, 20, 20);
        doc.Pages.Add(p1);
    }

    return View();
}
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.DiagramView
    MindFusion.Diagramming.Mvc.TabbedDiagramView

See Also

TabbedDiagramView Members
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView Members

Represents a DiagramView that can display DiagramDocument objects. The individual diagram pages can be activated through the tabs in the tray located at one of the view sides.

The following tables list the members exposed by the TabbedDiagramView type.

Public Constructors

	Name	Description

	TabbedDiagramView	Overloaded.

Public Properties

	Name	Description
	AddScriptReferences	Gets or sets a value indicating whether the control should render references to the client scripts. (Inherited from DiagramView.)
	AllowInplaceEdit	Gets or sets a value indicating whether in-place editing of the text of objects is enabled. (Inherited from DiagramView.)
	AllowReorder	Gets or sets a value indicating whether tabs can be rearranged interactively.
	AutoClasses	Gets the registered client types with JSON templates. (Inherited from DiagramView.)
	AutoScroll	Gets or sets a value indicating whether auto scrolling of the document area is enabled. (Inherited from DiagramView.)
	Behavior	Gets or sets a value indicating how the control responds to user actions. (Inherited from DiagramView.)
	ClientScriptLocation	Gets or sets the URL to the MindFusion.Diagramming.js file. (Inherited from DiagramView.)
	Converters	Gets the registered custom converters. (Inherited from DiagramView.)
	CustomLinkType	Gets the type of a MindFusion.Diagramming.DiagramLink derived class whose instance should be created when a user starts drawing. (Inherited from DiagramView.)
	CustomNodeType	Gets the type of a MindFusion.Diagramming.DiagramNode derived class whose instance should be created when a user starts drawing. (Inherited from DiagramView.)
	DefaultControlTemplate	Gets or sets the HTML string representing the DOM content, rendered by default in newly created ControlNode objects. (Inherited from DiagramView.)
	DelKeyAction	Gets or sets a value indicating the action that should be performed when the user hits the DELETE key. (Inherited from DiagramView.)
	Diagram	Gets or sets the Diagram object displayed by this view.
	Document	Gets or sets the DiagramDocument object displayed by this view.
	Enabled	Gets or sets a value indicating whether handling of mouse events is enabled. (Inherited from DiagramView.)
	EnableMultiTouch	Gets or sets a value indicating whether the control handles DOM pointer events to implement multi-touch interactions, such as zooming, node rotation or simultaneous drawing of multiple diagram items. (Inherited from DiagramView.)
	FallbackContent	Gets or sets the HTML to be rendered as the canvas' fallback content. (Inherited from DiagramView.)
	HandleTouchHitDistance	Gets or sets maximum distance between touch position and adjustment handle at which the handle can still be grabbed to modify its item. (Inherited from DiagramView.)
	Height	Gets or sets the height of the view. (Inherited from DiagramView.)
	Id	Gets or sets the id of the view. (Inherited from DiagramView.)
	ItemTypes	Gets the registered item types. (Inherited from DiagramView.)
	LicenseKey	Gets or sets the license key of the control. (Inherited from DiagramView.)
	MagnifierEnabled	Gets or sets a value indicating whether the magnifier tool is currently enabled. (Inherited from DiagramView.)

	MagnifierFactor	Gets or sets the zoom factor of the magnifier tool. (Inherited from DiagramView.)
	MagnifierFrameColor	Gets or sets the color of the magnifier frame. (Inherited from DiagramView.)
	MagnifierFrameThickness	Gets or sets the thickness of the magnifier frame, expressed in pixels. (Inherited from DiagramView.)
	MagnifierHeight	Gets or sets the height of the magnifier tool, expressed in pixels. (Inherited from DiagramView.)
	MagnifierSecondaryFrameColor	Gets or sets the secondary color of the magnifier frame. (Inherited from DiagramView.)
	MagnifierShading	Gets or sets a value indicating whether to enhance the effect of a magnifier lense by using a gradient shading. (Inherited from DiagramView.)
	MagnifierShape	Gets or sets the shape of the magnifier tool. (Inherited from DiagramView.)
	MagnifierWidth	Gets or sets the width of the magnifier tool, expressed in pixels. (Inherited from DiagramView.)
	MaxJsonLength	Gets or sets the maximum length of JSON strings that are accepted by internal JavaScriptSerializer instances. (Inherited from DiagramView.)
	MinVisibleFontSize	Gets or sets a threshold value that hides text if scaled font sizes become smaller. (Inherited from DiagramView.)
	ModificationStart	Gets or sets how users can start modifying diagram items. (Inherited from DiagramView.)
	MouseWheelAction	Gets or sets what action to perform when users rotate the mouse wheel. (Inherited from DiagramView.)
	MultiTouchDraw	Gets or sets a value indicating whether the control allows drawing multiple items simultaneously. (Inherited from DiagramView.)
	MultiTouchModify	Gets or sets a value indicating whether multi-touch modifications are enabled. (Inherited from DiagramView.)
	MultiTouchZoom	Gets or sets a value indicating whether multi-touch zoom is enabled. (Inherited from DiagramView.)
	RecursionLimit	Gets or sets the limit for constraining the number of object levels to process by internal JavaScriptSerializer instances. (Inherited from DiagramView.)
	ScrollLeft	Gets or sets the view's X scroll position. (Inherited from DiagramView.)
	ScrollTop	Gets or sets the view's Y scroll position. (Inherited from DiagramView.)
	SelectedIndex	Gets or sets the index of the currently selected tab page.
	ShapeLibraryLocation	The location of a shape library file containing custom shape definitions. It can be set to a full URL of the file, or a relative file path. (Inherited from DiagramView.)
	ShowAddButton	Gets or sets a value indicating whether to display the 'Add Tab' button.
	ShowNavigationButtons	Gets or sets a value indicating whether to show tab navigation buttons.
	ShowTabCloseButtons	Gets or sets a value indicating whether to display a close button for each tab.
	TabAlignment	Gets or sets the tab alignment.
	TabsSize	Gets or sets the height of the tab strip.
	ThemeId	Gets or sets the theme of the control.

	TooltipDelay	Gets or sets the delay before a tooltip is shown. (Inherited from DiagramView.)
	VirtualScroll	Gets or sets a value indicating whether virtual scroll mode is enabled. (Inherited from DiagramView.)
	Width	Gets or sets the width of the view. (Inherited from DiagramView.)
	ZoomFactor	Gets or sets the zoom of the view. (Inherited from DiagramView.)

Public Methods

	Name	Description
	AnimatedLayoutCompletedScript	Sets the name of a JavaScript function that should be invoked when an animated layout completes. (Inherited from DiagramView.)
	CellTextEditedScript	Sets the name of a JavaScript function that should be invoked when the user edits the text of a table cell. (Inherited from DiagramView.)
	CellTextEditingScript	Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a table cell. (Inherited from DiagramView.)
	ClickedScript	Sets the name of a JavaScript function that should be invoked when the user clicks the diagram at a point where no items are located. (Inherited from DiagramView.)
	ContainerChildAddedScript	Sets the name of a JavaScript function that should be invoked when a child node is added to a container. (Inherited from DiagramView.)
	ContainerChildAddingScript	Sets the name of a JavaScript function that should be invoked before a child node is added to a container. (Inherited from DiagramView.)
	ContainerChildRemovedScript	Sets the name of a JavaScript function that should be invoked when a child node is removed from a container. (Inherited from DiagramView.)
	ContainerChildRemovingScript	Sets the name of a JavaScript function that should be invoked before a child node is removed from a container. (Inherited from DiagramView.)
	ContainerFoldedScript	Sets the name of a JavaScript function that should be invoked when the user folds a container by clicking the arrow-up button in the container's caption area. (Inherited from DiagramView.)
	ContainerUnfoldedScript	Sets the name of a JavaScript function that should be invoked when the user unfolds a container by clicking the arrow-down button in the container's caption area. (Inherited from DiagramView.)
	ControlLoadedScript	Sets the name of a JavaScript function that should be invoked when the control is loaded on the client side. (Inherited from DiagramView.)
	CreateEditControlScript	Sets the name of a JavaScript function that should be invoked when when the inplace-edit control is created. (Inherited from DiagramView.)
	DataLoadedScript	Sets the name of a JavaScript function that should be invoked when the diagram data is loaded. (Inherited from DiagramView.)
	DeserializeTagScript	Sets the name of a JavaScript function that should be invoked when the Tag or Id of a diagram item must be deserialized from XML format. (Inherited from DiagramView.)
	DiagramToView	Overloaded. Transforms points from the document coordinate space to the client coordinate space. (Inherited from DiagramView.)
	DrawAdjustmentHandlesScript	Sets the name of a JavaScript function that should be invoked when item's adjustment handles must be custom drawn. (Inherited from DiagramView.)
	EnterInplaceEditScript	Sets the name of a JavaScript function that should be invoked when when entering inplace-edit mode. (Inherited from DiagramView.)
	FromJson	Overloaded.
	FromRequest	Creates an instance of TabbedDiagramView.

	HeaderResizedScript	Sets the name of a JavaScript function that should be invoked when the user finished resizing a header. (Inherited from DiagramView.)
	HeaderStartResizingScript	Sets the name of a JavaScript function that should be invoked when the user starts resizing a header. (Inherited from DiagramView.)
	HitTestAdjustmentHandlesScript	Sets the name of a JavaScript function that should be invoked when custom hit testing of item's adjustment handles must be performed. (Inherited from DiagramView.)
	HyperlinkClickedScript	Sets the name of a JavaScript function that should be invoked when the user clicks a hyperlink inside a node. (Inherited from DiagramView.)
	Init	Renders the TabbedDiagramView's initializing scripts on a web page under the specified function name.
	InitializeLinkScript	Sets the name of a JavaScript function that should be invoked when the user starts drawing a new link, just after the DiagramLink instance is created. (Inherited from DiagramView.)
	InitializeNodeScript	Sets the name of a JavaScript function that should be invoked when the user starts drawing a new node, just after the DiagramNode instance is created. (Inherited from DiagramView.)
	ItemAddedScript	Sets the name of a JavaScript function that should be invoked when an item is added to the diagram. (Inherited from DiagramView.)
	ItemRemovedScript	Sets the name of a JavaScript function that should be invoked when an item is removed from the diagram. (Inherited from DiagramView.)
	LaneGridCellTextEditedScript	Sets the name of a JavaScript function that should be invoked hen the used edits the text of a grid cell. (Inherited from DiagramView.)
	LeaveInplaceEditScript	Sets the name of a JavaScript function that should be invoked when when leaving inplace-edit mode. (Inherited from DiagramView.)
	LinkClickedScript	Sets the name of a JavaScript function that should be invoked when the users clicks a link. (Inherited from DiagramView.)
	LinkCreatedScript	Sets the name of a JavaScript function that should be invoked when the user draws a new link. (Inherited from DiagramView.)
	LinkCreatingScript	Sets the name of a JavaScript function that should be invoked while the user is drawing a new link. (Inherited from DiagramView.)
	LinkDeletedScript	Sets the name of a JavaScript function that should be invoked when a link is deleted, either programmatically or by the user. (Inherited from DiagramView.)
	LinkDeletingScript	Sets the name of a JavaScript function that should be invoked when the user tries to delete a link. (Inherited from DiagramView.)
	LinkDeselectedScript	Sets the name of a JavaScript function that should be invoked when a link is deselected, either programmatically or by the user. (Inherited from DiagramView.)
	LinkDoubleClickedScript	Sets the name of a JavaScript function that should be invoked when the users double-clicks a link. (Inherited from DiagramView.)
	LinkModifiedScript	Sets the name of a JavaScript function that should be invoked when the user moves any control point of a link. (Inherited from DiagramView.)
	LinkModifyingScript	Sets the name of a JavaScript function that should be invoked while the user is modifying a link. (Inherited from DiagramView.)
	LinkPastedScript	Sets the name of a JavaScript function that should be invoked when a diagram link is pasted from clipboard. (Inherited from DiagramView.)
	LinkPointedScript	Sets the name of a JavaScript function that should be invoked when the mouse pointer hovers over a link. (Inherited from DiagramView.)
	LinkSelectedScript	Sets the name of a JavaScript function that should be invoked when a link is selected, either programmatically or by the user. (Inherited from DiagramView.)

	LinkSelectingScript	Sets the name of a JavaScript function that should be invoked when the user tries to select a link. (Inherited from DiagramView.)
	LinkTextEditedScript	Sets the name of a JavaScript function that should be invoked when the user edits the text of a link. (Inherited from DiagramView.)
	LinkTextEditingScript	Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a link. (Inherited from DiagramView.)
	LoadFromJson	Overloaded. (Inherited from DiagramView.)
	MouseWheelScript	Sets the name of a JavaScript function that should be invoked when the user rotates the mouse wheel while the cursor is positioned over the diagram surface. (Inherited from DiagramView.)
	NodeClickedScript	Sets the name of a JavaScript function that should be invoked when the users clicks a node. (Inherited from DiagramView.)
	NodeCreatedScript	Sets the name of a JavaScript function that should be invoked when the user draws a new node. (Inherited from DiagramView.)
	NodeCreatingScript	Sets the name of a JavaScript function that should be invoked while the user is drawing a new node. (Inherited from DiagramView.)
	NodeDeletedScript	Sets the name of a JavaScript function that should be invoked when a node is deleted, either programmatically or by the user. (Inherited from DiagramView.)
	NodeDeletingScript	Sets the name of a JavaScript function that should be invoked when the user tries to delete a node. (Inherited from DiagramView.)
	NodeDeselectedScript	Sets the name of a JavaScript function that should be invoked when a node is deselected, either programmatically or by the user. (Inherited from DiagramView.)
	NodeDomCreatedScript	Sets the name of a JavaScript function that should be invoked when the content of a ControlNode is added to the page DOM. (Inherited from DiagramView.)
	NodeDoubleClickedScript	Sets the name of a JavaScript function that should be invoked when the users double-clicks a node. (Inherited from DiagramView.)
	NodeModifiedScript	Sets the name of a JavaScript function that should be invoked when the user moves or resizes a node. (Inherited from DiagramView.)
	NodeModifyingScript	Sets the name of a JavaScript function that should be invoked while the user is moving or resizing a node. (Inherited from DiagramView.)
	NodePastedScript	Sets the name of a JavaScript function that should be invoked when a diagram node is pasted from clipboard. (Inherited from DiagramView.)
	NodePointedScript	Sets the name of a JavaScript function that should be invoked when the mouse pointer hovers over a node. (Inherited from DiagramView.)
	NodeSelectedScript	Sets the name of a JavaScript function that should be invoked when a node is selected, either programmatically or by the user. (Inherited from DiagramView.)
	NodeSelectingScript	Sets the name of a JavaScript function that should be invoked when the user tries to select a node. (Inherited from DiagramView.)
	NodeTextEditedScript	Sets the name of a JavaScript function that should be invoked when the user edits the text of a node. (Inherited from DiagramView.)
	NodeTextEditingScript	Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a node. (Inherited from DiagramView.)
	PageAddedScript	Sets the name of a JavaScript function that should be invoked when the a page has been added to a TabbedDiagramView control.
	PageAddingScript	Sets the name of a JavaScript function that should be invoked before a page is added to a TabbedDiagramView control.

	PageChangedScript	Sets the name of a JavaScript function that should be invoked when the selected page in a TabbedDiagramView control has changed.
	PageChangingScript	Sets the name of a JavaScript function that should be invoked before the selected page in a TabbedDiagramView control is changed.
	PageRemovedScript	Sets the name of a JavaScript function that should be invoked when the a page has been removed from a TabbedDiagramView control.
	PageRemovingScript	Sets the name of a JavaScript function that should be invoked before a page is removed from a TabbedDiagramView control.
	RegisterConverters	Registers custom converters used for custom item types serialization. (Inherited from DiagramView.)
	RegisterItemType	Overloaded. (Inherited from DiagramView.)
	RegisterSession	Overloaded. (Inherited from DiagramView.)
	Render	Renders the **DiagramView** on a web page. (Inherited from DiagramView.)
	RepaintScript	Sets the name of a JavaScript function that should be invoked when the diagram has been repainted. (Inherited from DiagramView.)
	SaveToJson	Serializes the TabbedDiagramView's data to JSON.
	SelectionChangedScript	Sets the name of a JavaScript function that should be invoked when the diagram selection has changed. (Inherited from DiagramView.)
	SelectionModifyingScript	Sets the name of a JavaScript function that should be invoked while the user is moving or resizing multiple items. (Inherited from DiagramView.)
	SerializeTagScript	Sets the name of a JavaScript function that should be invoked when the Tag or Id of a diagram item must be serialized in XML format. (Inherited from DiagramView.)
	SetAddScriptReferences	Sets a value indicating whether the control should render references to the client scripts. (Inherited from DiagramView.)
	SetAllowInplaceEdit	Sets a value indicating whether in-place editing of the text of objects is enabled. (Inherited from DiagramView.)
	SetAllowReorder	Sets a value indicating whether tabs can be rearranged interactively.
	SetAutoScroll	Sets a value indicating whether auto scrolling of the document area is enabled. (Inherited from DiagramView.)
	SetBehavior	Sets a value indicating how the control responds to user actions. (Inherited from DiagramView.)
	SetCustomLinkType	Sets the type of a MindFusion.Diagramming.**DiagramLink** derived class whose instance should be created when a user starts drawing. (Inherited from DiagramView.)
	SetCustomNodeType	Sets the type of a MindFusion.Diagramming.**DiagramNode** derived class whose instance should be created when a user starts drawing. (Inherited from DiagramView.)
	SetDelKeyAction	Sets a value indicating the action that should be performed when the user hits the Del key. (Inherited from DiagramView.)
	SetEnabled	Sets a value indicating whether handling of mouse events is enabled. (Inherited from DiagramView.)
	SetEnableMultiTouch	Sets a value indicating whethr the control handles DOM pointer events to implement multi-touch interactions, such as zooming, node rotation or simultaneous drawing of multiple diagram items. (Inherited from DiagramView.)
	SetFallbackContent	Sets the HTML to be rendered as the canvas' fallback content. (Inherited from DiagramView.)
	SetHeight	Sets the height of the view. (Inherited from DiagramView.)

	SetMagnifierEnabled	Sets a value indicating whether the magnifier tool is currently enabled. (Inherited from DiagramView.)
	SetMagnifierFactor	Sets the zoom factor of the magnifier tool. (Inherited from DiagramView.)
	SetMagnifierFrameColor	Sets the color of the magnifier frame. (Inherited from DiagramView.)
	SetMagnifierFrameThickness	Sets the thickness of the magnifier frame, expressed in pixels. (Inherited from DiagramView.)
	SetMagnifierHeight	Sets the height of the magnifier tool, expressed in pixels. (Inherited from DiagramView.)
	SetMagnifierSecondaryFrameColor	Sets the secondary color of the magnifier frame. (Inherited from DiagramView.)
	SetMagnifierShading	Sets a value indicating whether to enhance the effect of a magnifier lense by using a gradient shading. (Inherited from DiagramView.)
	SetMagnifierShape	Sets the shape of the magnifier tool. (Inherited from DiagramView.)
	SetMagnifierWidth	Sets the width of the magnifier tool, expressed in pixels. (Inherited from DiagramView.)
	SetMaxJsonLength	Sets the maximum length of JSON strings that are accepted by internal JavaScriptSerializer instances. (Inherited from DiagramView.)
	SetMinVisibleFontSize	Sets a threshold value that hides text if scaled font sizes become smaller. (Inherited from DiagramView.)
	SetModificationStart	Sets how users can start modifying diagram items. (Inherited from DiagramView.)
	SetMouseWheelAction	Sets a value indicating what action to perform when users rotate the mouse wheel. (Inherited from DiagramView.)
	SetMultiTouchDraw	Sets a value indicating whether the control allows drawing multiple items simultaneously. (Inherited from DiagramView.)
	SetMultiTouchModify	Sets a value indicating whether multi-touch modifications are enabled. (Inherited from DiagramView.)
	SetMultiTouchZoom	Sets a value indicating whether multi-touch zoom is enabled. (Inherited from DiagramView.)
	SetRecursionLimit	Sets the limit for constraining the number of object levels to process by internal JavaScriptSerializer instances. (Inherited from DiagramView.)
	SetScrollLeft	Sets the view's X scroll position. (Inherited from DiagramView.)
	SetScrollTop	Sets the view's Y scroll position. (Inherited from DiagramView.)
	SetShowAddButton	Sets a value indicating whether to display the 'Add Tab' button.
	SetShowNavigationButtons	Sets a value indicating whether to show tab navigation buttons.
	SetShowScrollbars	Sets a value indicating whether the scrollbars are visible. (Inherited from DiagramView.)
	SetShowTabCloseButtons	Sets a value indicating whether to display a close button for each tab.
	SetTabAlignment	Sets the tab alignment.
	SetTabsSize	Sets the height of the tab strip.
	SetTooltipDelay	Sets the delay before a tooltip is shown. (Inherited from DiagramView.)
	SetVirtualScroll	Sets a value indicating whether virtual scroll mode is enabled. (Inherited from DiagramView.)
	SetWidth	Sets the width of the view. (Inherited from DiagramView.)
	SetZoomFactor	Sets the zoom of the view. (Inherited from DiagramView.)

	SizeChangedScript	Sets the name of a JavaScript function that should be invoked when the size of the diagram canvas has changed. (Inherited from DiagramView.)
	TableColumnResizedScript	Sets the name of a JavaScript function that should be invoked when the user finishes resizing a column in a table interactively. (Inherited from DiagramView.)
	TableColumnResizingScript	Sets the name of a JavaScript function that should be invoked when the user resizes a column in a table interactively. (Inherited from DiagramView.)
	TableRowResizedScript	Sets the name of a JavaScript function that should be invoked when the user finishes resizing a row in a table interactively. (Inherited from DiagramView.)
	TableRowResizingScript	Sets the name of a JavaScript function that should be invoked when the user resizes a row in a table interactively. (Inherited from DiagramView.)
	ToJson	Serializes the DiagramView's data to JSON.
	ToJson	Serializes the DiagramView's data to JSON. (Inherited from DiagramView.)
	TreeCollapsedScript	Sets the name of a JavaScript function that should be invoked when a user collapses a tree branch by clicking the [+] button of an Expandable node. (Inherited from DiagramView.)
	TreeExpandedScript	Sets the name of a JavaScript function that should be invoked when a user expands a tree branch by clicking the [+] button of an Expandable node. (Inherited from DiagramView.)
	TreeItemTextEditedScript	Sets the name of a JavaScript function that should be invoked when the user edits the text of a tree view item. (Inherited from DiagramView.)
	TreeItemTextEditingScript	Sets the name of a JavaScript function that should be invoked when the user starts to edit the text of a tree view item. (Inherited from DiagramView.)
	ViewToDiagram	Overloaded. (Inherited from DiagramView.)

Protected Methods

	Name	Description
	GetAttributes	(Inherited from DiagramView.)
	LoadViewProps	
	SaveViewProps	

See Also

TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView Constructor

Overload List

Name	Description
TabbedDiagramView ()	Creates an instance of TabbedDiagramView.
TabbedDiagramView (String)	Creates an instance of TabbedDiagramView.
TabbedDiagramView (String, DiagramDocument)	Creates an instance of TabbedDiagramView.

See Also

TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView Constructor ()

Creates an instance of TabbedDiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

TabbedDiagramView Constructor Overload List
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView Constructor (String)

Creates an instance of TabbedDiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView (
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String _
)
```

Parameters

id
The Id of the TabbedDiagramView

See Also

TabbedDiagramView Constructor Overload List
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView Constructor (String, DiagramDocument)

Creates an instance of TabbedDiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView (
    string id,
    DiagramDocument document
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String, _
    document As DiagramDocument _
)
```

Parameters

id
The Id of the TabbedDiagramView

document
The DiagramDocument displayed by the view

See Also

TabbedDiagramView Constructor Overload List
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView Methods

TabbedDiagramView.FromJson Method

Overload List

Name	Description
TabbedDiagramView.FromJson (String)	Creates a Diagram object from JSON.
TabbedDiagramView.FromJson (String, String)	Creates a Diagram object from JSON.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.FromJson Method (String)

Creates a Diagram object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public new static TabbedDiagramView FromJson (
    string json
```

```
)
```

Visual Basic Copy Code

```
Public Shadows Shared Function FromJson( _
    json As String _
) As TabbedDiagramView
```

Parameters

json
The JSON string

Return Value

See Also

FromJson Method Overload List
TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.FromJson Method (String, String)

Creates a Diagram object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public new static TabbedDiagramView FromJson (
    string id,
    string json
)
```

Visual Basic Copy Code

```
Public Shadows Shared Function FromJson( _
    id As String, _
    json As String _
) As TabbedDiagramView
```

Parameters

id
The Id of the TabbedDiagramView

json
The JSON string

Return Value

See Also

FromJson Method Overload List
TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.FromRequest Method

Creates an instance of TabbedDiagramView.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public new static TabbedDiagramView FromRequest (
    string id,
    HttpRequest request
)
```

Visual Basic Copy Code

```
Public Shadows Shared Function FromRequest( _
    id As String, _
    request As HttpRequest _
) As TabbedDiagramView
```

Parameters

id

The Id of the TabbedDiagramView

request

An HttpRequest object that contains the json data for the view.

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.Init Method

Renders the TabbedDiagramView's initializing scripts on a web page under the specified function name.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override string Init (
    string functionName
)
```

Visual Basic Copy Code

```
Public Overrides Function Init( _
    functionName As String _
) As String
```

Parameters

functionName

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.LoadViewProps Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
protected override void LoadViewProps (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub LoadViewProps( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

context

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.PageAddedScript Method

Sets the name of a JavaScript function that should be invoked when the a page has been added to a TabbedDiagramView control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView PageAddedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function PageAddedScript( _
    value As String _
) As TabbedDiagramView
```

Parameters

value
The name of the function

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.PageAddingScript Method

Sets the name of a JavaScript function that should be invoked before a page is added to a TabbedDiagramView control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView PageAddingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function PageAddingScript( _
    value As String _
) As TabbedDiagramView
```

Parameters

value
The name of the function

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.PageChangedScript Method

Sets the name of a JavaScript function that should be invoked when the selected page in a TabbedDiagramView control has changed.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView PageChangedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function PageChangedScript( _
    value As String _
) As TabbedDiagramView
```

Parameters

value
The name of the function

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.PageChangingScript Method

Sets the name of a JavaScript function that should be invoked before the selected page in a TabbedDiagramView control is changed.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView PageChangingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function PageChangingScript( _
    value As String _
) As TabbedDiagramView
```

Parameters

value
The name of the function

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.PageRemovedScript Method

Sets the name of a JavaScript function that should be invoked when the a page has been removed from a TabbedDiagramView control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView PageRemovedScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function PageRemovedScript( _
    value As String _
) As TabbedDiagramView
```

Parameters

value
The name of the function

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.PageRemovingScript Method

Sets the name of a JavaScript function that should be invoked before a page is removed from a TabbedDiagramView control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView PageRemovingScript (
    string value
)
```

Visual Basic Copy Code

```
Public Function PageRemovingScript( _
    value As String _
) As TabbedDiagramView
```

Parameters

value
The name of the function

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.SaveToJson Method

Serializes the TabbedDiagramView's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override string SaveToJson ()
```

Visual Basic Copy Code

```
Public Overrides Function SaveToJson () As String
```

Return Value

The JSON string.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.SaveViewProps Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
protected override void SaveViewProps (
    JsonObject obj,
    JsonPersistContext context
)
```

Visual Basic Copy Code

```
Protected Overrides Sub SaveViewProps( _
    obj As JsonObject, _
    context As JsonPersistContext _
)
```

Parameters

obj

context

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.SetAllowReorder Method

Sets a value indicating whether tabs can be rearranged interactively.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView SetAllowReorder (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAllowReorder( _
    value As Boolean _
) As TabbedDiagramView
```

Parameters

value

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.SetShowAddButton Method

Sets a value indicating whether to display the 'Add Tab' button.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView SetShowAddButton (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetShowAddButton( _
    value As Boolean _
) As TabbedDiagramView
```

Parameters

value

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.SetShowNavigationButtons Method

Sets a value indicating whether to show tab navigation buttons.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView SetShowNavigationButtons (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetShowNavigationButtons( _
    value As Boolean _
) As TabbedDiagramView
```

Parameters

value

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.SetShowTabCloseButtons Method

Sets a value indicating whether to display a close button for each tab.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView SetShowTabCloseButtons (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetShowTabCloseButtons( _
    value As Boolean _
) As TabbedDiagramView
```

Parameters

value

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.SetTabAlignment Method

Sets the tab alignment.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView SetTabAlignment (
    TabAlignment value
)
```

Visual Basic Copy Code

```
Public Function SetTabAlignment( _
    value As TabAlignment _
) As TabbedDiagramView
```

Parameters

value

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.SetTabsSize Method

Sets the height of the tab strip.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView SetTabsSize (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetTabsSize( _
    value As Integer _
) As TabbedDiagramView
```

Parameters

value

Return Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.ToJson Method

Serializes the DiagramView's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public new static string ToJson (
    TabbedDiagramView view
)
```

Visual Basic Copy Code

```
Public Shadows Shared Function ToJson( _
    view As TabbedDiagramView _
) As String
```

Parameters

view

Return Value

The JSON string.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView Properties

TabbedDiagramView.AllowReorder Property

Gets or sets a value indicating whether tabs can be rearranged interactively.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AllowReorder { get; set; }
```

Visual Basic Copy Code

```
Public Property AllowReorder As Boolean
```

Property Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.Diagram Property

Gets or sets the Diagram object displayed by this view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public new Diagram Diagram { get; set; }
```

Visual Basic Copy Code

```
Public Shadows Property Diagram As Diagram
```

Property Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.Document Property

Gets or sets the DiagramDocument object displayed by this view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public DiagramDocument Document { get; set; }
```

Visual Basic Copy Code

```
Public Property Document As DiagramDocument
```

Property Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.SelectedIndex Property

Gets or sets the index of the currently selected tab page.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int SelectedIndex { get; set; }
```

Visual Basic Copy Code

```
Public Property SelectedIndex As Integer
```

Property Value

An integer index specifying the current page.

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.ShowAddButton Property

Gets or sets a value indicating whether to display the 'Add Tab' button.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool ShowAddButton { get; set; }
```

Visual Basic Copy Code

```
Public Property ShowAddButton As Boolean
```

Property Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.ShowNavigationButtons Property

Gets or sets a value indicating whether to show tab navigation buttons.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool ShowNavigationButtons { get; set; }
```

Visual Basic Copy Code

```
Public Property ShowNavigationButtons As Boolean
```

Property Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.ShowTabCloseButtons Property

Gets or sets a value indicating whether to display a close button for each tab.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool ShowTabCloseButtons { get; set; }
```

Visual Basic Copy Code

```
Public Property ShowTabCloseButtons As Boolean
```

Property Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.TabAlignment Property

Gets or sets the tab alignment.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabAlignment TabAlignment { get; set; }
```

Visual Basic Copy Code

```
Public Property TabAlignment As TabAlignment
```

Property Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.TabsSize Property

Gets or sets the height of the tab strip.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int TabsSize { get; set; }
```

Visual Basic Copy Code

```
Public Property TabsSize As Integer
```

Property Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramView.ThemeId Property

Gets or sets the theme of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string ThemeId { get; set; }
```

Visual Basic Copy Code

```
Public Property ThemeId As String
```

Property Value

See Also

TabbedDiagramView Members
TabbedDiagramView Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions Class

Contains extension methods of TabbedDiagramView.

Syntax

C# Copy Code

```
public sealed class TabbedDiagramViewExtensions
```

Visual Basic Copy Code

```
Public NotInheritable Class TabbedDiagramViewExtensions
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.TabbedDiagramViewExtensions

See Also

TabbedDiagramViewExtensions Members
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions Members

Contains extension methods of TabbedDiagramView.

The following tables list the members exposed by the TabbedDiagramViewExtensions type.

Public Methods

	Name	Description
	TabbedDiagramView	Overloaded.

See Also

TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions Methods

TabbedDiagramViewExtensions.TabbedDiagramView Method

Overload List

Name	Description
TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, String, DiagramDocument)	Returns a TabbedDiagramView by using the specified HTML helper.
TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, String, DiagramDocument, IDictionary(String, Object))	Returns a TabbedDiagramView by using the specified HTML helper.
TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, String, DiagramDocument, Object)	Returns a TabbedDiagramView by using the specified HTML helper.
TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, String, DiagramDocument, String)	Returns a TabbedDiagramView by using the specified HTML helper.

TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, String, DiagramDocument, String, IDictionary(String, Object))	Returns a TabbedDiagramView by using the specified HTML helper.
TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, String, DiagramDocument, String, Object)	Returns a TabbedDiagramView by using the specified HTML helper.
TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, TabbedDiagramView)	Returns a TabbedDiagramView by using the specified HTML helper.
TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, TabbedDiagramView, IDictionary(String, Object))	Returns a TabbedDiagramView by using the specified HTML helper.
TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, TabbedDiagramView, Object)	Returns a TabbedDiagramView by using the specified HTML helper.
TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, TabbedDiagramView, String)	Returns a TabbedDiagramView by using the specified HTML helper.
TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, TabbedDiagramView, String, IDictionary(String, Object))	Returns a TabbedDiagramView by using the specified HTML helper.
TabbedDiagramViewExtensions.TabbedDiagramView (IHtmlHelper, TabbedDiagramView, String, Object)	Returns a TabbedDiagramView by using the specified HTML helper.

See Also

TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method (IHtmlHelper, String, DiagramDocument)

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    string id,
    DiagramDocument document
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    id As String, _
    document As DiagramDocument _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the TabbedDiagramView

document
The DiagramDocument displayed by the view

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method (IHtmlHelper, String, DiagramDocument, IDictionary(String, Object))

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    string id,
    DiagramDocument document,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    id As String, _
    document As DiagramDocument, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the TabbedDiagramView

document
The DiagramDocument displayed by the view

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method (IHtmlHelper, String, DiagramDocument, Object)

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    string id,
    DiagramDocument document,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    id As String, _
    document As DiagramDocument, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the TabbedDiagramView

document
The DiagramDocument displayed by the view

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method (IHtmlHelper, String, DiagramDocument, String)

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    string id,
    DiagramDocument document,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    id As String, _
```

```
    document As DiagramDocument, _
    initFunction As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the TabbedDiagramView

document
The DiagramDocument displayed by the view

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method (IHtmlHelper, String, DiagramDocument, String, IDictionary(String, Object))

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    string id,
    DiagramDocument document,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    id As String, _
    document As DiagramDocument, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the TabbedDiagramView

document
The DiagramDocument displayed by the view

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method (IHtmlHelper, String, DiagramDocument, String, Object)

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    string id,
    DiagramDocument document,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    id As String, _
    document As DiagramDocument, _
    initFunction As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the TabbedDiagramView

document
The DiagramDocument displayed by the view

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method (IHtmlHelper, TabbedDiagramView)

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    TabbedDiagramView tabbedDiagramView
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    tabbedDiagramView As TabbedDiagramView _
) As IHtmlContent
```

Parameters

helper
The HTML helper

tabbedDiagramView
The TabbedDiagramView

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method (IHtmlHelper, TabbedDiagramView, IDictionary(String, Object))

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    TabbedDiagramView tabbedDiagramView,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    tabbedDiagramView As TabbedDiagramView, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

tabbedDiagramView
The TabbedDiagramView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method (IHtmlHelper, TabbedDiagramView, Object)

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    TabbedDiagramView tabbedDiagramView,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    tabbedDiagramView As TabbedDiagramView, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

tabbedDiagramView
The TabbedDiagramView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method

(IHtmlHelper, TabbedDiagramView, String)

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    TabbedDiagramView tabbedDiagramView,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    tabbedDiagramView As TabbedDiagramView, _
    initFunction As String _
) As IHtmlContent
```

Parameters

helper
The HTML helper

tabbedDiagramView
The TabbedDiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method (IHtmlHelper, TabbedDiagramView, String, IDictionary(String, Object))

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    TabbedDiagramView tabbedDiagramView,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    tabbedDiagramView As TabbedDiagramView, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

tabbedDiagramView
The TabbedDiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewExtensions.TabbedDiagramView Method (IHtmlHelper, TabbedDiagramView, String, Object)

Returns a TabbedDiagramView by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent TabbedDiagramView (
    IHtmlHelper helper,
    TabbedDiagramView tabbedDiagramView,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function TabbedDiagramView( _
    helper As IHtmlHelper, _
    tabbedDiagramView As TabbedDiagramView, _
    initFunction As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

tabbedDiagramView
The TabbedDiagramView

initFunction
The name of the function that will contain the initializing scripts for the view.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

TabbedDiagramView Method Overload List
TabbedDiagramViewExtensions Members
TabbedDiagramViewExtensions Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewTagHelper Class

Tag helper of TabbedDiagramView.

Syntax

C# Copy Code

```
public class TabbedDiagramViewTagHelper : TagHelper
```

Visual Basic Copy Code

```
Public Class TabbedDiagramViewTagHelper
    Inherits TagHelper
```

Remarks

This class implements the <mindfusion-tabbeddiagramview> tag.

Inheritance Hierarchy

System.Object
  Microsoft.AspNetCore.Razor.TagHelpers.TagHelper
    MindFusion.Diagramming.Mvc.TabbedDiagramViewTagHelper

See Also

TabbedDiagramViewTagHelper Members
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewTagHelper Members

Tag helper of TabbedDiagramView.

The following tables list the members exposed by the TabbedDiagramViewTagHelper type.

Public Constructors

	Name	Description
	TabbedDiagramViewTagHelper	

Public Properties

	Name	Description
	Component	
	Id	
	Style	

Public Methods

	Name	Description

	Process	

See Also

TabbedDiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewTagHelper Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramViewTagHelper ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

TabbedDiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewTagHelper Methods

TabbedDiagramViewTagHelper.Process Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override void Process (
    TagHelperContext context,
    TagHelperOutput output
)
```

Visual Basic Copy Code

```
Public Overrides Sub Process( _
    context As TagHelperContext, _
    output As TagHelperOutput _
)
```

Parameters

context

output

See Also

TabbedDiagramViewTagHelper Members
TabbedDiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewTagHelper Properties

TabbedDiagramViewTagHelper.Component Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TabbedDiagramView Component { get; set; }
```

Visual Basic Copy Code

```
Public Property Component As TabbedDiagramView
```

Property Value

See Also

TabbedDiagramViewTagHelper Members
TabbedDiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewTagHelper.Id Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

TabbedDiagramViewTagHelper Members
TabbedDiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

TabbedDiagramViewTagHelper.Style Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As String
```

Property Value

See Also

TabbedDiagramViewTagHelper Members
TabbedDiagramViewTagHelper Class
MindFusion.Diagramming.Mvc Namespace

TableNodeConverter Class

Serializes TableNode instances.

Syntax

C# Copy Code

```
public class TableNodeConverter : DiagramNodeConverter
```

Visual Basic Copy Code

```
Public Class TableNodeConverter
    Inherits DiagramNodeConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter
    MindFusion.Diagramming.Mvc.DiagramItemConverter
      MindFusion.Diagramming.Mvc.DiagramNodeConverter
        MindFusion.Diagramming.Mvc.TableNodeConverter

See Also

TableNodeConverter Members
MindFusion.Diagramming.Mvc Namespace

TableNodeConverter Members

Serializes TableNode instances.

The following tables list the members exposed by the TableNodeConverter type.

Public Constructors

	Name	Description
	TableNodeConverter	Creates a new instance of the TableNodeConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.

	Serialize	Serialize override.

See Also

TableNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

TableNodeConverter Constructor

Creates a new instance of the TableNodeConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TableNodeConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

TableNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

TableNodeConverter Methods

TableNodeConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

TableNodeConverter Members
TableNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

TableNodeConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

TableNodeConverter Members
TableNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

TableNodeConverter Properties

TableNodeConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

TableNodeConverter Members
TableNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

TickPosition Enumeration

Specifies the position of trackbar ticks.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public enum TickPosition
```

Visual Basic Copy Code

```
Public Enum TickPosition
```

Members

	Member name	Description
	None	There are no ticks rendered.
	Left	Ticks are rendered on the left side of the trackbar.
	Right	Ticks are rendered on the right side of the trackbar.
	Both	Ticks are rendered on both sides of the trackbar.

See Also

MindFusion.Diagramming.Mvc Namespace

TreeViewNodeConverter Class

Serializes TreeViewNode instances.

Syntax

C# Copy Code

```
public class TreeViewNodeConverter : DiagramNodeConverter
```

Visual Basic Copy Code

```
Public Class TreeViewNodeConverter
    Inherits DiagramNodeConverter
```

Inheritance Hierarchy

System.Object
  System.Web.Script.Serialization.JavaScriptConverter

MindFusion.Diagramming.Mvc.DiagramItemConverter
MindFusion.Diagramming.Mvc.DiagramNodeConverter
MindFusion.Diagramming.Mvc.TreeViewNodeConverter

See Also

TreeViewNodeConverter Members
MindFusion.Diagramming.Mvc Namespace

TreeViewNodeConverter Members

Serializes TreeViewNode instances.

The following tables list the members exposed by the TreeViewNodeConverter type.

Public Constructors

	Name	Description
	TreeViewNodeConverter	Creates a new instance of the TreeViewNodeConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

TreeViewNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

TreeViewNodeConverter Constructor

Creates a new instance of the TreeViewNodeConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TreeViewNodeConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

TreeViewNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

TreeViewNodeConverter Methods

TreeViewNodeConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

TreeViewNodeConverter Members
TreeViewNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

TreeViewNodeConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

TreeViewNodeConverter Members
TreeViewNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

TreeViewNodeConverter Properties

TreeViewNodeConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

TreeViewNodeConverter Members
TreeViewNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

VideoNodeConverter Class

Serializes VideoNode instances.

Syntax

C# Copy Code

```
public class VideoNodeConverter : DiagramNodeConverter
```

Visual Basic Copy Code

```
Public Class VideoNodeConverter
    Inherits DiagramNodeConverter
```

Inheritance Hierarchy

System.Object
System.Web.Script.Serialization.JavaScriptConverter
MindFusion.Diagramming.Mvc.DiagramItemConverter
MindFusion.Diagramming.Mvc.DiagramNodeConverter
MindFusion.Diagramming.Mvc.VideoNodeConverter

See Also

VideoNodeConverter Members
MindFusion.Diagramming.Mvc Namespace

VideoNodeConverter Members

Serializes VideoNode instances.

The following tables list the members exposed by the VideoNodeConverter type.

Public Constructors

	Name	Description
	VideoNodeConverter	Creates a new instance of the VideoNodeConverter class.

Public Properties

	Name	Description
	SupportedTypes	SupportedTypes override.

Public Methods

	Name	Description
	Deserialize	Deserialize override.
	Serialize	Serialize override.

See Also

VideoNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

VideoNodeConverter Constructor

Creates a new instance of the VideoNodeConverter class.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public VideoNodeConverter (
    DiagramView view
)
```

Visual Basic Copy Code

```
Public New ( _
    view As DiagramView _
)
```

Parameters

view

See Also

VideoNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

VideoNodeConverter Methods

VideoNodeConverter.Deserialize Method

Deserialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override Object Deserialize (
    IDictionary<string, Object> dictionary,
    Type type,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Deserialize( _
    dictionary As IDictionary(Of String, Object), _
    type As Type, _
    serializer As JavaScriptSerializer _
) As Object
```

Parameters

dictionary

type

serializer

Return Value

See Also

VideoNodeConverter Members
VideoNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

VideoNodeConverter.Serialize Method

Serialize override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IDictionary<string, Object> Serialize (
    Object obj,
    JavaScriptSerializer serializer
)
```

Visual Basic Copy Code

```
Public Overrides Function Serialize( _
    obj As Object, _
    serializer As JavaScriptSerializer _
) As IDictionary(Of String, Object)
```

Parameters

obj

serializer

Return Value

See Also

VideoNodeConverter Members
VideoNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

VideoNodeConverter Properties

VideoNodeConverter.SupportedTypes Property

SupportedTypes override.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override IEnumerable<Type> SupportedTypes { get; }
```

Visual Basic Copy Code

```
Public Overrides ReadOnly Property SupportedTypes As IEnumerable(Of Type)
```

Property Value

See Also

VideoNodeConverter Members
VideoNodeConverter Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl Class

ZoomControl lets users zoom and pan a target DiagramView interactively.

Syntax

C# Copy Code

```
public class ZoomControl
```

Visual Basic Copy Code

```
Public Class ZoomControl
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.ZoomControl

See Also

ZoomControl Members
MindFusion.Diagramming.Mvc Namespace

ZoomControl Members

ZoomControl lets users zoom and pan a target DiagramView interactively.

The following tables list the members exposed by the **ZoomControl** type.

Public Constructors

	Name	Description
	ZoomControl	Overloaded.

Public Properties

	Name	Description
	ActiveColor	Gets or sets the color used to render depressed buttons.
	AddScriptReferences	Gets or sets a value indicating whether the control should render references to the client scripts.
	BackColor	Gets or sets the background color of the control.
	BorderColor	Gets or sets the color of ZoomControl elements' borders.
	ClientScriptLocation	Gets or sets the URL to the MindFusion.Diagramming.js file.
	CornerRadius	Gets or sets the corner radius of rounded child elements.
	Enabled	Gets or sets a value indicating whether handling of mouse events is enabled.
	Fill	Gets or sets the brush used to fill the **ZoomControl** elements.
	Height	Gets or sets the height of the control.
	Id	Gets or sets the id of the view.
	InnerColor	Gets or sets the color of plus, minus and arrow icons.
	MaxZoomFactor	The maximum zoom level allowed to set through this control.
	MinZoomFactor	The minimum zoom level allowed to set through this control.
	Padding	Gets or sets the padding of the control's contents.
	ScrollStep	Gets or sets the scroll offset added when users click the pan arrows.
	ShadowColor	Gets or sets the shadow color.
	ShowLabel	Gets or sets a value indicating whether the label that shows the current zoom level should be visible.
	SnapToZoomStep	Gets or sets a value indicating whether the trackbar should snap to ZoomStep values when dragged.
	TargetId	Gets or sets the id of the target view.
	TextColor	Gets or sets the brush of the the label that shows the current zoom level.
	TickPosition	Gets or sets the current tick position of the trackbar.
	Width	Gets or sets the width of the control.
	ZoomFactor	Gets or sets the current zoom level.

	ZoomStep	Gets or sets the amount by which to change zoom level when + and - buttons are clicked.

Public Methods

	Name	Description
	FromJson	Overloaded.
	FromRequest	Overloaded.
	Init	Renders the **ZoomControl's** initializing scripts on a web page under the specified function name.
	LoadFromJson	Loads the specified JSON data.
	Render	Renders the **ZoomControl** on a web page.
	SaveToJson	Serializes the ZoomControl's data to JSON.
	SetActiveColor	Sets the color used to render depressed buttons.
	SetAddScriptReferences	Sets a value indicating whether the control should render references to the client scripts.
	SetBackColor	Sets the background color of the control.
	SetBorderColor	Sets the color of **ZoomControl** elements' borders.
	SetCornerRadius	Sets the corner radius of rounded child elements.
	SetEnabled	Sets a value indicating whether handling of mouse events is enabled.
	SetFill	Sets the brush used to fill the **ZoomControl** elements.
	SetHeight	Sets the height of the view.
	SetInnerColor	Sets the color of plus, minus and arrow icons.
	SetMaxZoomFactor	Sets the maximum zoom level allowed to set through this control.
	SetMinZoomFactor	Sets the minimum zoom level allowed to set through this control.
	SetPadding	Sets the padding of the control's contents.
	SetScrollStep	Sets the scroll offset added when users click the pan arrows.
	SetShadowColor	Sets the shadow color.
	SetShowLabel	Sets a value indicating whether the label that shows the current zoom level should be visible.
	SetSnapToZoomStep	Sets a value indicating whether the trackbar should snap to **ZoomStep** values when dragged.
	SetTargetId	Sets the id of the target view.
	SetTextColor	Sets the brush of the the label that shows the current zoom level.
	SetTickPosition	Sets the current tick position of the trackbar.
	SetWidth	Sets the width of the view.
	SetZoomFactor	Sets the id of the target view.
	SetZoomStep	Sets the amount by which to change zoom level when + and - buttons are clicked.
	ToJson	Serializes the ZoomControl's data to JSON.

See Also

ZoomControl Constructor

Overload List

Name	Description
ZoomControl ()	Creates an instance of ZoomControl.
ZoomControl (String)	Creates an instance of **ZoomControl**.
ZoomControl (String, String)	Creates an instance of **ZoomControl** for a target view.

See Also

ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl Constructor ()

Creates an instance of ZoomControl.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ZoomControl Constructor Overload List
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl Constructor (String)

Creates an instance of ZoomControl.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl (
    string id
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String _
```

```
)
```

Parameters

id

See Also

ZoomControl Constructor Overload List
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl Constructor (String, String)

Creates an instance of ZoomControl for a target view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl (
    string id,
    string targetId
)
```

Visual Basic Copy Code

```
Public New ( _
    id As String, _
    targetId As String _
)
```

Parameters

id
The Id of the ZoomControl

targetId
The Id of the target DiagramView

See Also

ZoomControl Constructor Overload List
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl Methods

ZoomControl.FromJson Method

Overload List

Name	Description
ZoomControl.FromJson (String)	Creates a ZoomControl object from JSON.
ZoomControl.FromJson (String, String)	Creates a ZoomControl object from JSON.

See Also

ZoomControl.FromJson Method (String)

Creates a ZoomControl object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static ZoomControl FromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    json As String _
) As ZoomControl
```

Parameters

json
The JSON string

Return Value

See Also

FromJson Method Overload List
ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.FromJson Method (String, String)

Creates a ZoomControl object from JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static ZoomControl FromJson (
    string id,
    string json
)
```

Visual Basic Copy Code

```
Public Shared Function FromJson( _
    id As String, _
    json As String _
) As ZoomControl
```

Parameters

id
The Id of the ZoomControl

json

The JSON string

Return Value

See Also

FromJson Method Overload List
ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.FromRequest Method

Overload List

Name	Description
ZoomControl.FromRequest (String, HttpRequest)	Creates an instance of ZoomControl.
ZoomControl.FromRequest (String, HttpRequestBase)	Creates an instance of ZoomControl.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.FromRequest Method (String, HttpRequest)

Creates an instance of ZoomControl.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static ZoomControl FromRequest (
    string id,
    HttpRequest request
)
```

Visual Basic Copy Code

```
Public Shared Function FromRequest( _
    id As String, _
    request As HttpRequest _
) As ZoomControl
```

Parameters

id
The Id of the ZoomControl

request
An HttpRequest object that contains the json data for the view.

Return Value

See Also

FromRequest Method Overload List
ZoomControl Members

ZoomControl.FromRequest Method (String, HttpRequestBase)

Creates an instance of ZoomControl.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static ZoomControl FromRequest (
    string id,
    HttpRequestBase request
)
```

Visual Basic Copy Code

```
Public Shared Function FromRequest( _
    id As String, _
    request As HttpRequestBase _
) As ZoomControl
```

Parameters

id
The Id of the ZoomControl

request
An HttpRequest object that contains the json data for the view.

Return Value

See Also

FromRequest Method Overload List
ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.Init Method

Renders the ZoomControl's initializing scripts on a web page under the specified function name.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Init (
    string functionName
)
```

Visual Basic Copy Code

```
Public Function Init( _
    functionName As String _
) As String
```

Parameters

functionName

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.LoadFromJson Method

Loads the specified JSON data.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public void LoadFromJson (
    string json
)
```

Visual Basic Copy Code

```
Public Sub LoadFromJson( _
    json As String _
)
```

Parameters

json
The JSON string

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.Render Method

Renders the ZoomControl on a web page.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Render ()
```

Visual Basic Copy Code

```
Public Function Render () As String
```

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SaveToJson Method

Serializes the ZoomControl's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string SaveToJson ()
```

Visual Basic Copy Code

```
Public Function SaveToJson () As String
```

Return Value

The JSON string.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetActiveColor Method

Sets the color used to render depressed buttons.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetActiveColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetActiveColor( _
    value As Color _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetAddScriptReferences Method

Sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetAddScriptReferences (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetAddScriptReferences( _
    value As Boolean _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetBackColor Method

Sets the background color of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetBackColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetBackColor( _
    value As Color _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetBorderColor Method

Sets the color of ZoomControl elements' borders.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetBorderColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetBorderColor( _
    value As Color _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetCornerRadius Method

Sets the corner radius of rounded child elements.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetCornerRadius (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetCornerRadius( _
    value As Integer _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetEnabled Method

Sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetEnabled (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetEnabled( _
    value As Boolean _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetFill Method

Sets the brush used to fill the ZoomControl elements.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetFill (
    Brush value
)
```

Visual Basic Copy Code

```
Public Function SetFill( _
    value As Brush _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetHeight Method

Sets the height of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetHeight (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetHeight( _
    value As Integer _
) As ZoomControl
```

Parameters

value
The height of the view in pixels

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetInnerColor Method

Sets the color of plus, minus and arrow icons.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetInnerColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetInnerColor( _
    value As Color _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetMaxZoomFactor Method

Sets the maximum zoom level allowed to set through this control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetMaxZoomFactor (
    double value
)
```

Visual Basic Copy Code

```
Public Function SetMaxZoomFactor( _
    value As Double _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetMinZoomFactor Method

Sets the minimum zoom level allowed to set through this control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetMinZoomFactor (
    double value
)
```

Visual Basic Copy Code

```
Public Function SetMinZoomFactor( _
    value As Double _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetPadding Method

Sets the padding of the control's contents.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetPadding (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetPadding( _
    value As Integer _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetScrollStep Method

Sets the scroll offset added when users click the pan arrows.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetScrollStep (
    double value
)
```

Visual Basic Copy Code

```
Public Function SetScrollStep( _
    value As Double _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetShadowColor Method

Sets the shadow color.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetShadowColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetShadowColor( _
    value As Color _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetShowLabel Method

Sets a value indicating whether the label that shows the current zoom level should be visible.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetShowLabel (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetShowLabel( _
    value As Boolean _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetSnapToZoomStep Method

Sets a value indicating whether the trackbar should snap to ZoomStep values when dragged.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetSnapToZoomStep (
    bool value
)
```

Visual Basic Copy Code

```
Public Function SetSnapToZoomStep( _
    value As Boolean _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetTargetId Method

Sets the id of the target view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetTargetId (
    string value
)
```

Visual Basic Copy Code

```
Public Function SetTargetId( _
    value As String _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetTextColor Method

Sets the brush of the the label that shows the current zoom level.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetTextColor (
    Color value
)
```

Visual Basic Copy Code

```
Public Function SetTextColor( _
    value As Color _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetTickPosition Method

Sets the current tick position of the trackbar.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetTickPosition (
    TickPosition value
)
```

Visual Basic Copy Code

```
Public Function SetTickPosition( _
    value As TickPosition _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetWidth Method

Sets the width of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetWidth (
    int value
)
```

Visual Basic Copy Code

```
Public Function SetWidth( _
    value As Integer _
) As ZoomControl
```

Parameters

value
The width of the view in pixels

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetZoomFactor Method

Sets the id of the target view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetZoomFactor (
    double value
)
```

Visual Basic Copy Code

```
Public Function SetZoomFactor( _
    value As Double _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SetZoomStep Method

Sets the amount by which to change zoom level when + and - buttons are clicked.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl SetZoomStep (
    double value
)
```

Visual Basic Copy Code

```
Public Function SetZoomStep( _
    value As Double _
) As ZoomControl
```

Parameters

value

Return Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.ToJson Method

Serializes the ZoomControl's data to JSON.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static string ToJson (
    ZoomControl view
)
```

Visual Basic Copy Code

```
Public Shared Function ToJson( _
    view As ZoomControl _
) As String
```

Parameters

view

Return Value

The JSON string.

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl Properties

ZoomControl.ActiveColor Property

Gets or sets the color used to render depressed buttons.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color ActiveColor { get; set; }
```

Visual Basic Copy Code

```
Public Property ActiveColor As Color
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.AddScriptReferences Property

Gets or sets a value indicating whether the control should render references to the client scripts.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool AddScriptReferences { get; set; }
```

Visual Basic Copy Code

```
Public Property AddScriptReferences As Boolean
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.BackColor Property

Gets or sets the background color of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color BackColor { get; set; }
```

Visual Basic Copy Code

```
Public Property BackColor As Color
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.BorderColor Property

Gets or sets the color of ZoomControl elements' borders.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color BorderColor { get; set; }
```

Visual Basic Copy Code

```
Public Property BorderColor As Color
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.ClientScriptLocation Property

Gets or sets the URL to the MindFusion.Diagramming.js file.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string ClientScriptLocation { get; set; }
```

Visual Basic Copy Code

```
Public Property ClientScriptLocation As String
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.CornerRadius Property

Gets or sets the corner radius of rounded child elements.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int CornerRadius { get; set; }
```

Visual Basic Copy Code

```
Public Property CornerRadius As Integer
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.Enabled Property

Gets or sets a value indicating whether handling of mouse events is enabled.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool Enabled { get; set; }
```

Visual Basic Copy Code

```
Public Property Enabled As Boolean
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.Fill Property

Gets or sets the brush used to fill the ZoomControl elements.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Brush Fill { get; set; }
```

Visual Basic Copy Code

```
Public Property Fill As Brush
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.Height Property

Gets or sets the height of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Height { get; set; }
```

Visual Basic Copy Code

```
Public Property Height As Integer
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.Id Property

Gets or sets the id of the view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.InnerColor Property

Gets or sets the color of plus, minus and arrow icons.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color InnerColor { get; set; }
```

Visual Basic Copy Code

```
Public Property InnerColor As Color
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.MaxZoomFactor Property

The maximum zoom level allowed to set through this control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public double MaxZoomFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property MaxZoomFactor As Double
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.MinZoomFactor Property

The minimum zoom level allowed to set through this control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public double MinZoomFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property MinZoomFactor As Double
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.Padding Property

Gets or sets the padding of the control's contents.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Padding { get; set; }
```

Visual Basic Copy Code

```
Public Property Padding As Integer
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.ScrollStep Property

Gets or sets the scroll offset added when users click the pan arrows.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public double ScrollStep { get; set; }
```

Visual Basic Copy Code

```
Public Property ScrollStep As Double
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.ShadowColor Property

Gets or sets the shadow color.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color ShadowColor { get; set; }
```

Visual Basic Copy Code

```
Public Property ShadowColor As Color
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.ShowLabel Property

Gets or sets a value indicating whether the label that shows the current zoom level should be visible.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool ShowLabel { get; set; }
```

Visual Basic Copy Code

```
Public Property ShowLabel As Boolean
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.SnapToZoomStep Property

Gets or sets a value indicating whether the trackbar should snap to ZoomStep values when dragged.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public bool SnapToZoomStep { get; set; }
```

Visual Basic Copy Code

```
Public Property SnapToZoomStep As Boolean
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.TargetId Property

Gets or sets the id of the target view.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string TargetId { get; set; }
```

Visual Basic Copy Code

```
Public Property TargetId As String
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.TextColor Property

Gets or sets the brush of the the label that shows the current zoom level.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public Color TextColor { get; set; }
```

Visual Basic Copy Code

```
Public Property TextColor As Color
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.TickPosition Property

Gets or sets the current tick position of the trackbar.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public TickPosition TickPosition { get; set; }
```

Visual Basic Copy Code

```
Public Property TickPosition As TickPosition
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.Width Property

Gets or sets the width of the control.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public int Width { get; set; }
```

Visual Basic Copy Code

```
Public Property Width As Integer
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.ZoomFactor Property

Gets or sets the current zoom level.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public double ZoomFactor { get; set; }
```

Visual Basic Copy Code

```
Public Property ZoomFactor As Double
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControl.ZoomStep Property

Gets or sets the amount by which to change zoom level when + and - buttons are clicked.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public double ZoomStep { get; set; }
```

Visual Basic Copy Code

```
Public Property ZoomStep As Double
```

Property Value

See Also

ZoomControl Members
ZoomControl Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions Class

Contains extension methods of ZoomControl.

Syntax

C# Copy Code

```
public sealed class ZoomControlExtensions
```

Visual Basic Copy Code

```
Public NotInheritable Class ZoomControlExtensions
```

Inheritance Hierarchy

System.Object
  MindFusion.Diagramming.Mvc.ZoomControlExtensions

See Also

ZoomControlExtensions Members
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions Members

Contains extension methods of ZoomControl.

The following tables list the members exposed by the ZoomControlExtensions type.

Public Methods

	Name	Description
	ZoomControl	Overloaded.

See Also

ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions Methods

ZoomControlExtensions.ZoomControl Method

Overload List

Name	Description
ZoomControlExtensions.ZoomControl (IHtmlHelper, String, String)	Returns a ZoomControl by using the specified HTML helper.
ZoomControlExtensions.ZoomControl (IHtmlHelper, String, String, IDictionary(String, Object))	Returns a **ZoomControl** by using the specified HTML helper.
ZoomControlExtensions.ZoomControl (IHtmlHelper, String, String, Object)	Returns a **ZoomControl** by using the specified HTML helper.
ZoomControlExtensions.ZoomControl (IHtmlHelper, String, String, String)	Returns a **ZoomControl** by using the specified HTML helper.
ZoomControlExtensions.ZoomControl (IHtmlHelper, String, String, String, IDictionary(String, Object))	Returns a **ZoomControl** by using the specified HTML helper.
ZoomControlExtensions.ZoomControl (IHtmlHelper, String, String, String, Object)	Returns a **ZoomControl** by using the specified HTML helper.
ZoomControlExtensions.ZoomControl (IHtmlHelper, ZoomControl)	Returns a **ZoomControl** by using the specified HTML helper.
ZoomControlExtensions.ZoomControl (IHtmlHelper, ZoomControl, IDictionary(String, Object))	Returns a **ZoomControl** by using the specified HTML helper.
ZoomControlExtensions.ZoomControl (IHtmlHelper, ZoomControl, Object)	Returns a **ZoomControl** by using the specified HTML helper.
ZoomControlExtensions.ZoomControl (IHtmlHelper, ZoomControl, String)	Returns a **ZoomControl** by using the specified HTML helper.
ZoomControlExtensions.ZoomControl (IHtmlHelper, ZoomControl, String, IDictionary(String, Object))	Returns a **ZoomControl** by using the specified HTML helper.
ZoomControlExtensions.ZoomControl (IHtmlHelper, ZoomControl, String, Object)	Returns a **ZoomControl** by using the specified HTML helper.

See Also

ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, String, String)

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString ZoomControl (
    HtmlHelper helper,
    string id,
    string diagramViewId
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As HtmlHelper, _
    id As String, _
    diagramViewId As String _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

id
The Id of the ZoomControl

diagramViewId
The Id of the DiagramView

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, String, String, IDictionary(String, Object))

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ZoomControl (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ZoomControl

diagramViewId
The Id of the DiagramView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, String, String, Object)

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ZoomControl (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ZoomControl

diagramViewId
The Id of the DiagramView

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, String, String, String)

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString ZoomControl (
    HtmlHelper helper,
    string id,
    string diagramViewId,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As HtmlHelper, _
    id As String, _
    diagramViewId As String, _
    initFunction As String _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

id
The Id of the ZoomControl

diagramViewId
The Id of the DiagramView

initFunction
The name of the function that will contain the initializing scripts for the control.

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, String, String, String, IDictionary(String, Object))

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ZoomControl (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ZoomControl

diagramViewId
The Id of the DiagramView

initFunction
The name of the function that will contain the initializing scripts for the control.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, String, String, String, Object)

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ZoomControl (
    IHtmlHelper helper,
    string id,
    string diagramViewId,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As IHtmlHelper, _
    id As String, _
    diagramViewId As String, _
    initFunction As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

id
The Id of the ZoomControl

diagramViewId
The Id of the DiagramView

initFunction
The name of the function that will contain the initializing scripts for the control.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, ZoomControl)

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString ZoomControl (
    HtmlHelper helper,
    ZoomControl control
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As HtmlHelper, _
    control As ZoomControl _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

control
The ZoomControl

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, ZoomControl, IDictionary(String, Object))

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ZoomControl (
    IHtmlHelper helper,
    ZoomControl control,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As IHtmlHelper, _
    control As ZoomControl, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

control
The ZoomControl

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, ZoomControl, Object)

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ZoomControl (
    IHtmlHelper helper,
    ZoomControl control,
```

```
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As IHtmlHelper, _
    control As ZoomControl, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

control
The ZoomControl

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, ZoomControl, String)

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static MvcHtmlString ZoomControl (
    HtmlHelper helper,
    ZoomControl control,
    string initFunction
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As HtmlHelper, _
    control As ZoomControl, _
    initFunction As String _
) As MvcHtmlString
```

Parameters

helper
The HTML helper

control
The ZoomControl

initFunction
The name of the function that will contain the initializing scripts for the control.

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, ZoomControl, String, IDictionary(String, Object))

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ZoomControl (
    IHtmlHelper helper,
    ZoomControl control,
    string initFunction,
    IDictionary<string, Object> htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As IHtmlHelper, _
    control As ZoomControl, _
    initFunction As String, _
    htmlAttributes As IDictionary(Of String, Object) _
) As IHtmlContent
```

Parameters

helper
The HTML helper

control
The ZoomControl

initFunction
The name of the function that will contain the initializing scripts for the control.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlExtensions.ZoomControl Method (IHtmlHelper, ZoomControl, String, Object)

Returns a ZoomControl by using the specified HTML helper.

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public static IHtmlContent ZoomControl (
    IHtmlHelper helper,
    ZoomControl control,
    string initFunction,
    Object htmlAttributes
)
```

Visual Basic Copy Code

```
Public Shared Function ZoomControl( _
    helper As IHtmlHelper, _
    control As ZoomControl, _
    initFunction As String, _
    htmlAttributes As Object _
) As IHtmlContent
```

Parameters

helper
The HTML helper

control
The ZoomControl

initFunction
The name of the function that will contain the initializing scripts for the control.

htmlAttributes
An object containing the HTML attributes for the control.

Return Value

See Also

ZoomControl Method Overload List
ZoomControlExtensions Members
ZoomControlExtensions Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlTagHelper Class

Tag helper of ZoomControl.

Syntax

C# Copy Code

```
public class ZoomControlTagHelper : TagHelper
```

Visual Basic Copy Code

```
Public Class ZoomControlTagHelper
    Inherits TagHelper
```

Remarks

This class implements the <mindfusion-zoomcontrol> tag.

Inheritance Hierarchy

System.Object
  Microsoft.AspNetCore.Razor.TagHelpers.TagHelper
    MindFusion.Diagramming.Mvc.ZoomControlTagHelper

See Also

ZoomControlTagHelper Members
MindFusion.Diagramming.Mvc Namespace

ZoomControlTagHelper Members

Tag helper of ZoomControl.

The following tables list the members exposed by the ZoomControlTagHelper type.

Public Constructors

	Name	Description
	ZoomControlTagHelper	

Public Properties

	Name	Description
	Component	
	Id	
	Style	

Public Methods

	Name	Description
	Process	

See Also

ZoomControlTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlTagHelper Constructor

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControlTagHelper ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

ZoomControlTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlTagHelper Methods

ZoomControlTagHelper.Process Method

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public override void Process (
    TagHelperContext context,
    TagHelperOutput output
)
```

Visual Basic Copy Code

```
Public Overrides Sub Process( _
    context As TagHelperContext, _
    output As TagHelperOutput _
)
```

Parameters

context

output

See Also

ZoomControlTagHelper Members
ZoomControlTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlTagHelper Properties

ZoomControlTagHelper.Component Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public ZoomControl Component { get; set; }
```

Visual Basic Copy Code

```
Public Property Component As ZoomControl
```

Property Value

See Also

ZoomControlTagHelper Members
ZoomControlTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlTagHelper.Id Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Id { get; set; }
```

Visual Basic Copy Code

```
Public Property Id As String
```

Property Value

See Also

ZoomControlTagHelper Members
ZoomControlTagHelper Class
MindFusion.Diagramming.Mvc Namespace

ZoomControlTagHelper.Style Property

Namespace: MindFusion.Diagramming.Mvc
Assembly: MindFusion.Diagramming.Mvc

Syntax

C# Copy Code

```
public string Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As String
```

Property Value

See Also

ZoomControlTagHelper Members
ZoomControlTagHelper Class
MindFusion.Diagramming.Mvc Namespace

MindFusion.Drawing Namespace

Brush Class

Brush objects are used to fill the interiors of geometric shapes. **Brush** is abstract base class inherited by SolidBrush, HatchBrush, TextureBrush, PathGradientBrush and LinearGradientBrush.

Syntax

C# Copy Code

```
public abstract class Brush
```

Visual Basic Copy Code

```
Public MustInherit Class Brush
```

Inheritance Hierarchy

System.Object
MindFusion.Drawing.Brush
MindFusion.Drawing.HatchBrush
MindFusion.Drawing.LinearGradientBrush
MindFusion.Drawing.PathGradientBrush
MindFusion.Drawing.SolidBrush
MindFusion.Drawing.TextureBrush

See Also

Brush Members
MindFusion.Drawing Namespace

Brush Members

Brush objects are used to fill the interiors of geometric shapes. **Brush** is abstract base class inherited by SolidBrush, HatchBrush, TextureBrush, PathGradientBrush and LinearGradientBrush.

The following tables list the members exposed by the **Brush** type.

Protected Constructors

	Name	Description
	Brush	Initializes a new instance of the **Brush** class.

Public Methods

	Name	Description
	Clone	Creates an exact copy of this brush.
	Equals	Overloaded.
	GetHashCode	Object.GethashCode override.
	ToColor	Returns a solid color approximating the brush.

See Also

Brush Class
MindFusion.Drawing Namespace

Brush Constructor

Initializes a new instance of the Brush class.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
protected Brush ()
```

Visual Basic Copy Code

```
Protected Sub New ()
```

See Also

Brush Class
MindFusion.Drawing Namespace

Brush Methods

Brush.Clone Method

Creates an exact copy of this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public abstract Brush Clone ()
```

Visual Basic Copy Code

```
Public MustOverride Function Clone () As Brush
```

Return Value

See Also

Brush Members
Brush Class
MindFusion.Drawing Namespace

Brush.Equals Method

Overload List

Name	Description
Brush.Equals (Brush)	Determines whether the specified Brush is equal to the current Brush.
Brush.Equals (Object)	Object.Equals override.

See Also

Brush Members
Brush Class
MindFusion.Drawing Namespace

Brush.Equals Method (Brush)

Determines whether the specified Brush is equal to the current Brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public abstract bool Equals (
    Brush other
)
```

Visual Basic Copy Code

```
Public MustOverride Function Equals( _
    other As Brush _
```

```
) As Boolean
```

Parameters

other
The Brush to compare with the current Brush.

Return Value

true if the specified Brush is equal to the current Brush; otherwise, false.

See Also

Equals Method Overload List
Brush Members
Brush Class
MindFusion.Drawing Namespace

Brush.Equals Method (Object)

Object.Equals override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override bool Equals (
    Object obj
)
```

Visual Basic Copy Code

```
Public Overrides Function Equals( _
    obj As Object _
) As Boolean
```

Parameters

obj
The object to compare with the current Brush.

Return Value

true if the specified object is equal to the current Brush; otherwise, false.

See Also

Equals Method Overload List
Brush Members
Brush Class
MindFusion.Drawing Namespace

Brush.GetHashCode Method

Object.GethashCode override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override int GetHashCode ()
```

Visual Basic Copy Code

```
Public Overrides Function GetHashCode () As Integer
```

Return Value

See Also

Brush Members
Brush Class
MindFusion.Drawing Namespace

Brush.ToColor Method

Returns a solid color approximating the brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public abstract Color ToColor ()
```

Visual Basic Copy Code

```
Public MustOverride Function ToColor () As Color
```

Return Value

See Also

Brush Members
Brush Class
MindFusion.Drawing Namespace

Brushes Class

Provides a set of predefined brushes.

Syntax

C# Copy Code

```
public sealed class Brushes
```

Visual Basic Copy Code

```
Public NotInheritable Class Brushes
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.Brushes

See Also

Brushes Members
MindFusion.Drawing Namespace

Brushes Members

Provides a set of predefined brushes.

The following tables list the members exposed by the Brushes type.

Public Fields

	Name	Description
	AliceBlue	A solid brush with the AntiqueWhite color.
	AntiqueWhite	A solid brush with the AntiqueWhite color.
	Aqua	A solid brush with the Aqua color.
	Aquamarine	A solid brush with the Aquamarine color.
	Azure	A solid brush with the Azure color.
	Beige	A solid brush with the Beige color.
	Bisque	A solid brush with the Bisque color.
	Black	A solid brush with the Black color.
	BlanchedAlmond	A solid brush with the BlanchedAlmond color.
	Blue	A solid brush with the Blue color.
	BlueViolet	A solid brush with the BlueViolet color.
	Brown	A solid brush with the Brown color.
	BurlyWood	A solid brush with the BurlyWood color.
	CadetBlue	A solid brush with the CadetBlue color.
	Chartreuse	A solid brush with the Chartreuse color.
	Chocolate	A solid brush with the Chocolate color.
	Coral	A solid brush with the Coral color.
	CornflowerBlue	A solid brush with the CornflowerBlue color.
	Cornsilk	A solid brush with the Cornsilk color.
	Crimson	A solid brush with the Crimson color.
	Cyan	A solid brush with the Cyan color.
	DarkBlue	A solid brush with the DarkBlue color.
	DarkCyan	A solid brush with the DarkCyan color.
	DarkGoldenrod	A solid brush with the DarkGoldenrod color.
	DarkGray	A solid brush with the DarkGray color.
	DarkGreen	A solid brush with the DarkGreen color.
	DarkKhaki	A solid brush with the DarkKhaki color.
	DarkMagenta	A solid brush with the DarkMagenta color.
	DarkOliveGreen	A solid brush with the DarkOliveGreen color.
	DarkOrange	A solid brush with the DarkOrange color.
	DarkOrchid	A solid brush with the DarkOrchid color.
	DarkRed	A solid brush with the DarkRed color.
	DarkSalmon	A solid brush with the DarkSalmon color.
	DarkSeaGreen	A solid brush with the DarkSeaGreen color.

	DarkSlateBlue	A solid brush with the DarkSlateBlue color.
	DarkSlateGray	A solid brush with the DarkSlateGray color.
	DarkTurquoise	A solid brush with the DarkTurquoise color.
	DarkViolet	A solid brush with the DarkViolet color.
	DeepPink	A solid brush with the DeepPink color.
	DeepSkyBlue	A solid brush with the DeepSkyBlue color.
	DimGray	A solid brush with the DimGray color.
	DodgerBlue	A solid brush with the DodgerBlue color.
	Firebrick	A solid brush with the Firebrick color.
	FloralWhite	A solid brush with the FloralWhite color.
	ForestGreen	A solid brush with the ForestGreen color.
	Fuchsia	A solid brush with the Fuchsia color.
	Gainsboro	A solid brush with the Gainsboro color.
	GhostWhite	A solid brush with the GhostWhite color.
	Gold	A solid brush with the Gold color.
	Goldenrod	A solid brush with the Goldenrod color.
	Gray	A solid brush with the Gray color.
	Green	A solid brush with the Green color.
	GreenYellow	A solid brush with the GreenYellow color.
	Honeydew	A solid brush with the Honeydew color.
	HotPink	A solid brush with the HotPink color.
	IndianRed	A solid brush with the IndianRed color.
	Indigo	A solid brush with the Indigo color.
	Ivory	A solid brush with the Ivory color.
	Khaki	A solid brush with the Khaki color.
	Lavender	A solid brush with the Lavender color.
	LavenderBlush	A solid brush with the LavenderBlush color.
	LawnGreen	A solid brush with the LawnGreen color.
	LemonChiffon	A solid brush with the LemonChiffon color.
	LightBlue	A solid brush with the LightBlue color.
	LightCoral	A solid brush with the LightCoral color.
	LightCyan	A solid brush with the LightCyan color.
	LightGoldenrodYellow	A solid brush with the LightGoldenrodYellow color.
	LightGray	A solid brush with the LightGray color.
	LightGreen	A solid brush with the LightGreen color.
	LightPink	A solid brush with the LightPink color.
	LightSalmon	A solid brush with the LightSalmon color.

	LightSeaGreen	A solid brush with the LightSeaGreen color.
	LightSkyBlue	A solid brush with the LightSkyBlue color.
	LightSlateGray	A solid brush with the LightSlateGray color.
	LightSteelBlue	A solid brush with the LightSteelBlue color.
	LightYellow	A solid brush with the LightYellow color.
	Lime	A solid brush with the Lime color.
	LimeGreen	A solid brush with the LimeGreen color.
	Linen	A solid brush with the Linen color.
	Magenta	A solid brush with the Magenta color.
	Maroon	A solid brush with the Maroon color.
	MediumAquamarine	A solid brush with the MediumAquamarine color.
	MediumBlue	A solid brush with the MediumBlue color.
	MediumOrchid	A solid brush with the MediumOrchid color.
	MediumPurple	A solid brush with the MediumPurple color.
	MediumSeaGreen	A solid brush with the MediumSeaGreen color.
	MediumSlateBlue	A solid brush with the MediumSlateBlue color.
	MediumSpringGreen	A solid brush with the MediumSpringGreen color.
	MediumTurquoise	A solid brush with the MediumTurquoise color.
	MediumVioletRed	A solid brush with the MediumVioletRed color.
	MidnightBlue	A solid brush with the MidnightBlue color.
	MintCream	A solid brush with the MintCream color.
	MistyRose	A solid brush with the MistyRose color.
	Moccasin	A solid brush with the Moccasin color.
	NavajoWhite	A solid brush with the NavajoWhite color.
	Navy	A solid brush with the Navy color.
	OldLace	A solid brush with the OldLace color.
	Olive	A solid brush with the Olive color.
	OliveDrab	A solid brush with the OliveDrab color.
	Orange	A solid brush with the Orange color.
	OrangeRed	A solid brush with the OrangeRed color.
	Orchid	A solid brush with the Orchid color.
	PaleGoldenrod	A solid brush with the PaleGoldenrod color.
	PaleGreen	A solid brush with the PaleGreen color.
	PaleTurquoise	A solid brush with the PaleTurquoise color.
	PaleVioletRed	A solid brush with the PaleVioletRed color.
	PapayaWhip	A solid brush with the PapayaWhip color.
	PeachPuff	A solid brush with the PeachPuff color.

	Peru	A solid brush with the Peru color.
	Pink	A solid brush with the Pink color.
	Plum	A solid brush with the Plum color.
	PowderBlue	A solid brush with the PowderBlue color.
	Purple	A solid brush with the Purple color.
	Red	A solid brush with the Red color.
	RosyBrown	A solid brush with the RosyBrown color.
	RoyalBlue	A solid brush with the RoyalBlue color.
	SaddleBrown	A solid brush with the SaddleBrown color.
	Salmon	A solid brush with the Salmon color.
	SandyBrown	A solid brush with the SandyBrown color.
	SeaGreen	A solid brush with the SeaGreen color.
	SeaShell	A solid brush with the SeaShell color.
	Sienna	A solid brush with the Sienna color.
	Silver	A solid brush with the Silver color.
	SkyBlue	A solid brush with the SkyBlue color.
	SlateBlue	A solid brush with the SlateBlue color.
	SlateGray	A solid brush with the SlateGray color.
	Snow	A solid brush with the Snow color.
	SpringGreen	A solid brush with the SpringGreen color.
	SteelBlue	A solid brush with the SteelBlue color.
	Tan	A solid brush with the Tan color.
	Teal	A solid brush with the Teal color.
	Thistle	A solid brush with the Thistle color.
	Tomato	A solid brush with the Tomato color.
	Transparent	A solid brush with the Transparent color.
	Turquoise	A solid brush with the Turquoise color.
	Violet	A solid brush with the Violet color.
	Wheat	A solid brush with the Wheat color.
	White	A solid brush with the White color.
	WhiteSmoke	A solid brush with the WhiteSmoke color.
	Yellow	A solid brush with the Yellow color.
	YellowGreen	A solid brush with the YellowGreen color.

See Also

Brushes Class
MindFusion.Drawing Namespace

Brushes Fields

Brushes.AliceBlue Field

A solid brush with the AntiqueWhite color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush AliceBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly AliceBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.AntiqueWhite Field

A solid brush with the AntiqueWhite color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush AntiqueWhite
```

Visual Basic Copy Code

```
Public Shared ReadOnly AntiqueWhite As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Aqua Field

A solid brush with the Aqua color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Aqua
```

Visual Basic Copy Code

```
Public Shared ReadOnly Aqua As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Aquamarine Field

A solid brush with the Aquamarine color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Aquamarine
```

Visual Basic Copy Code

```
Public Shared ReadOnly Aquamarine As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Azure Field

A solid brush with the Azure color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Azure
```

Visual Basic Copy Code

```
Public Shared ReadOnly Azure As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Beige Field

A solid brush with the Beige color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Beige
```

Visual Basic Copy Code

```
Public Shared ReadOnly Beige As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Bisque Field

A solid brush with the Bisque color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Bisque
```

Visual Basic Copy Code

```
Public Shared ReadOnly Bisque As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Black Field

A solid brush with the Black color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Black
```

Visual Basic Copy Code

```
Public Shared ReadOnly Black As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.BlanchedAlmond Field

A solid brush with the BlanchedAlmond color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush BlanchedAlmond
```

Visual Basic Copy Code

```
Public Shared ReadOnly BlanchedAlmond As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Blue Field

A solid brush with the Blue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Blue
```

Visual Basic Copy Code

```
Public Shared ReadOnly Blue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.BlueViolet Field

A solid brush with the BlueViolet color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush BlueViolet
```

Visual Basic Copy Code

```
Public Shared ReadOnly BlueViolet As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Brown Field

A solid brush with the Brown color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Brown
```

Visual Basic Copy Code

```
Public Shared ReadOnly Brown As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.BurlyWood Field

A solid brush with the BurlyWood color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush BurlyWood
```

Visual Basic Copy Code

```
Public Shared ReadOnly BurlyWood As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.CadetBlue Field

A solid brush with the CadetBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush CadetBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly CadetBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Chartreuse Field

A solid brush with the Chartreuse color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Chartreuse
```

Visual Basic Copy Code

```
Public Shared ReadOnly Chartreuse As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Chocolate Field

A solid brush with the Chocolate color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Chocolate
```

Visual Basic Copy Code

```
Public Shared ReadOnly Chocolate As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Coral Field

A solid brush with the Coral color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Coral
```

Visual Basic Copy Code

```
Public Shared ReadOnly Coral As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.CornflowerBlue Field

A solid brush with the CornflowerBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush CornflowerBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly CornflowerBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Cornsilk Field

A solid brush with the Cornsilk color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Cornsilk
```

Visual Basic Copy Code

```
Public Shared ReadOnly Cornsilk As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Crimson Field

A solid brush with the Crimson color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Crimson
```

Visual Basic Copy Code

```
Public Shared ReadOnly Crimson As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Cyan Field

A solid brush with the Cyan color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Cyan
```

Visual Basic Copy Code

```
Public Shared ReadOnly Cyan As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkBlue Field

A solid brush with the DarkBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkCyan Field

A solid brush with the DarkCyan color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkCyan
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkCyan As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkGoldenrod Field

A solid brush with the DarkGoldenrod color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkGoldenrod
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkGoldenrod As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkGray Field

A solid brush with the DarkGray color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkGray
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkGray As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkGreen Field

A solid brush with the DarkGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkKhaki Field

A solid brush with the DarkKhaki color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkKhaki
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkKhaki As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkMagenta Field

A solid brush with the DarkMagenta color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkMagenta
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkMagenta As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkOliveGreen Field

A solid brush with the DarkOliveGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkOliveGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkOliveGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkOrange Field

A solid brush with the DarkOrange color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkOrange
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkOrange As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkOrchid Field

A solid brush with the DarkOrchid color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkOrchid
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkOrchid As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkRed Field

A solid brush with the DarkRed color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkRed
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkRed As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkSalmon Field

A solid brush with the DarkSalmon color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkSalmon
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkSalmon As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkSeaGreen Field

A solid brush with the DarkSeaGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkSeaGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkSeaGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkSlateBlue Field

A solid brush with the DarkSlateBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkSlateBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkSlateBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkSlateGray Field

A solid brush with the DarkSlateGray color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkSlateGray
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkSlateGray As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkTurquoise Field

A solid brush with the DarkTurquoise color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkTurquoise
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkTurquoise As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DarkViolet Field

A solid brush with the DarkViolet color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DarkViolet
```

Visual Basic Copy Code

```
Public Shared ReadOnly DarkViolet As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DeepPink Field

A solid brush with the DeepPink color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DeepPink
```

Visual Basic Copy Code

```
Public Shared ReadOnly DeepPink As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DeepSkyBlue Field

A solid brush with the DeepSkyBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DeepSkyBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly DeepSkyBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DimGray Field

A solid brush with the DimGray color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DimGray
```

Visual Basic Copy Code

```
Public Shared ReadOnly DimGray As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.DodgerBlue Field

A solid brush with the DodgerBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush DodgerBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly DodgerBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Firebrick Field

A solid brush with the Firebrick color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Firebrick
```

Visual Basic Copy Code

```
Public Shared ReadOnly Firebrick As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.FloralWhite Field

A solid brush with the FloralWhite color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush FloralWhite
```

Visual Basic Copy Code

```
Public Shared ReadOnly FloralWhite As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.ForestGreen Field

A solid brush with the ForestGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush ForestGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly ForestGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Fuchsia Field

A solid brush with the Fuchsia color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Fuchsia
```

Visual Basic Copy Code

```
Public Shared ReadOnly Fuchsia As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Gainsboro Field

A solid brush with the Gainsboro color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Gainsboro
```

Visual Basic Copy Code

```
Public Shared ReadOnly Gainsboro As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.GhostWhite Field

A solid brush with the GhostWhite color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush GhostWhite
```

Visual Basic Copy Code

```
Public Shared ReadOnly GhostWhite As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Gold Field

A solid brush with the Gold color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Gold
```

Visual Basic Copy Code

```
Public Shared ReadOnly Gold As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Goldenrod Field

A solid brush with the Goldenrod color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Goldenrod
```

Visual Basic Copy Code

```
Public Shared ReadOnly Goldenrod As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Gray Field

A solid brush with the Gray color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Gray
```

Visual Basic Copy Code

```
Public Shared ReadOnly Gray As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Green Field

A solid brush with the Green color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Green
```

Visual Basic Copy Code

```
Public Shared ReadOnly Green As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.GreenYellow Field

A solid brush with the GreenYellow color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush GreenYellow
```

Visual Basic Copy Code

```
Public Shared ReadOnly GreenYellow As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Honeydew Field

A solid brush with the Honeydew color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Honeydew
```

Visual Basic Copy Code

```
Public Shared ReadOnly Honeydew As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.HotPink Field

A solid brush with the HotPink color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush HotPink
```

Visual Basic Copy Code

```
Public Shared ReadOnly HotPink As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.IndianRed Field

A solid brush with the IndianRed color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush IndianRed
```

Visual Basic Copy Code

```
Public Shared ReadOnly IndianRed As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Indigo Field

A solid brush with the Indigo color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Indigo
```

Visual Basic Copy Code

```
Public Shared ReadOnly Indigo As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Ivory Field

A solid brush with the Ivory color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Ivory
```

Visual Basic Copy Code

```
Public Shared ReadOnly Ivory As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Khaki Field

A solid brush with the Khaki color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Khaki
```

Visual Basic Copy Code

```
Public Shared ReadOnly Khaki As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Lavender Field

A solid brush with the Lavender color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Lavender
```

Visual Basic Copy Code

```
Public Shared ReadOnly Lavender As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LavenderBlush Field

A solid brush with the LavenderBlush color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LavenderBlush
```

Visual Basic Copy Code

```
Public Shared ReadOnly LavenderBlush As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LawnGreen Field

A solid brush with the LawnGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LawnGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly LawnGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LemonChiffon Field

A solid brush with the LemonChiffon color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LemonChiffon
```

Visual Basic Copy Code

```
Public Shared ReadOnly LemonChiffon As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightBlue Field

A solid brush with the LightBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightCoral Field

A solid brush with the LightCoral color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightCoral
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightCoral As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightCyan Field

A solid brush with the LightCyan color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightCyan
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightCyan As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightGoldenrodYellow Field

A solid brush with the LightGoldenrodYellow color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightGoldenrodYellow
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightGoldenrodYellow As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightGray Field

A solid brush with the LightGray color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightGray
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightGray As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightGreen Field

A solid brush with the LightGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightPink Field

A solid brush with the LightPink color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightPink
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightPink As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightSalmon Field

A solid brush with the LightSalmon color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightSalmon
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightSalmon As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightSeaGreen Field

A solid brush with the LightSeaGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightSeaGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightSeaGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightSkyBlue Field

A solid brush with the LightSkyBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightSkyBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightSkyBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightSlateGray Field

A solid brush with the LightSlateGray color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightSlateGray
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightSlateGray As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightSteelBlue Field

A solid brush with the LightSteelBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightSteelBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightSteelBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LightYellow Field

A solid brush with the LightYellow color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LightYellow
```

Visual Basic Copy Code

```
Public Shared ReadOnly LightYellow As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Lime Field

A solid brush with the Lime color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Lime
```

Visual Basic Copy Code

```
Public Shared ReadOnly Lime As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.LimeGreen Field

A solid brush with the LimeGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush LimeGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly LimeGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Linen Field

A solid brush with the Linen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Linen
```

Visual Basic Copy Code

```
Public Shared ReadOnly Linen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Magenta Field

A solid brush with the Magenta color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Magenta
```

Visual Basic Copy Code

```
Public Shared ReadOnly Magenta As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Maroon Field

A solid brush with the Maroon color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Maroon
```

Visual Basic Copy Code

```
Public Shared ReadOnly Maroon As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MediumAquamarine Field

A solid brush with the MediumAquamarine color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MediumAquamarine
```

Visual Basic Copy Code

```
Public Shared ReadOnly MediumAquamarine As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MediumBlue Field

A solid brush with the MediumBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MediumBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly MediumBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MediumOrchid Field

A solid brush with the MediumOrchid color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MediumOrchid
```

Visual Basic Copy Code

```
Public Shared ReadOnly MediumOrchid As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MediumPurple Field

A solid brush with the MediumPurple color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MediumPurple
```

Visual Basic Copy Code

```
Public Shared ReadOnly MediumPurple As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MediumSeaGreen Field

A solid brush with the MediumSeaGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MediumSeaGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly MediumSeaGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MediumSlateBlue Field

A solid brush with the MediumSlateBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MediumSlateBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly MediumSlateBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MediumSpringGreen Field

A solid brush with the MediumSpringGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MediumSpringGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly MediumSpringGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MediumTurquoise Field

A solid brush with the MediumTurquoise color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MediumTurquoise
```

Visual Basic Copy Code

```
Public Shared ReadOnly MediumTurquoise As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MediumVioletRed Field

A solid brush with the MediumVioletRed color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MediumVioletRed
```

Visual Basic Copy Code

```
Public Shared ReadOnly MediumVioletRed As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MidnightBlue Field

A solid brush with the MidnightBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MidnightBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly MidnightBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MintCream Field

A solid brush with the MintCream color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MintCream
```

Visual Basic Copy Code

```
Public Shared ReadOnly MintCream As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.MistyRose Field

A solid brush with the MistyRose color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush MistyRose
```

Visual Basic Copy Code

```
Public Shared ReadOnly MistyRose As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Moccasin Field

A solid brush with the Moccasin color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Moccasin
```

Visual Basic Copy Code

```
Public Shared ReadOnly Moccasin As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.NavajoWhite Field

A solid brush with the NavajoWhite color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush NavajoWhite
```

Visual Basic Copy Code

```
Public Shared ReadOnly NavajoWhite As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Navy Field

A solid brush with the Navy color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Navy
```

Visual Basic Copy Code

```
Public Shared ReadOnly Navy As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.OldLace Field

A solid brush with the OldLace color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush OldLace
```

Visual Basic Copy Code

```
Public Shared ReadOnly OldLace As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Olive Field

A solid brush with the Olive color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Olive
```

Visual Basic Copy Code

```
Public Shared ReadOnly Olive As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.OliveDrab Field

A solid brush with the OliveDrab color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush OliveDrab
```

Visual Basic Copy Code

```
Public Shared ReadOnly OliveDrab As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Orange Field

A solid brush with the Orange color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Orange
```

Visual Basic Copy Code

```
Public Shared ReadOnly Orange As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.OrangeRed Field

A solid brush with the OrangeRed color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush OrangeRed
```

Visual Basic Copy Code

```
Public Shared ReadOnly OrangeRed As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Orchid Field

A solid brush with the Orchid color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Orchid
```

Visual Basic Copy Code

```
Public Shared ReadOnly Orchid As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.PaleGoldenrod Field

A solid brush with the PaleGoldenrod color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush PaleGoldenrod
```

Visual Basic Copy Code

```
Public Shared ReadOnly PaleGoldenrod As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.PaleGreen Field

A solid brush with the PaleGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush PaleGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly PaleGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.PaleTurquoise Field

A solid brush with the PaleTurquoise color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush PaleTurquoise
```

Visual Basic Copy Code

```
Public Shared ReadOnly PaleTurquoise As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.PaleVioletRed Field

A solid brush with the PaleVioletRed color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush PaleVioletRed
```

Visual Basic Copy Code

```
Public Shared ReadOnly PaleVioletRed As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.PapayaWhip Field

A solid brush with the PapayaWhip color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush PapayaWhip
```

Visual Basic Copy Code

```
Public Shared ReadOnly PapayaWhip As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.PeachPuff Field

A solid brush with the PeachPuff color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush PeachPuff
```

Visual Basic Copy Code

```
Public Shared ReadOnly PeachPuff As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Peru Field

A solid brush with the Peru color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Peru
```

Visual Basic Copy Code

```
Public Shared ReadOnly Peru As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Pink Field

A solid brush with the Pink color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Pink
```

Visual Basic Copy Code

```
Public Shared ReadOnly Pink As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Plum Field

A solid brush with the Plum color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Plum
```

Visual Basic Copy Code

```
Public Shared ReadOnly Plum As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.PowderBlue Field

A solid brush with the PowderBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush PowderBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly PowderBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Purple Field

A solid brush with the Purple color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Purple
```

Visual Basic Copy Code

```
Public Shared ReadOnly Purple As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Red Field

A solid brush with the Red color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Red
```

Visual Basic Copy Code

```
Public Shared ReadOnly Red As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.RosyBrown Field

A solid brush with the RosyBrown color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush RosyBrown
```

Visual Basic Copy Code

```
Public Shared ReadOnly RosyBrown As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.RoyalBlue Field

A solid brush with the RoyalBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush RoyalBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly RoyalBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.SaddleBrown Field

A solid brush with the SaddleBrown color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush SaddleBrown
```

Visual Basic Copy Code

```
Public Shared ReadOnly SaddleBrown As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Salmon Field

A solid brush with the Salmon color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Salmon
```

Visual Basic Copy Code

```
Public Shared ReadOnly Salmon As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.SandyBrown Field

A solid brush with the SandyBrown color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush SandyBrown
```

Visual Basic Copy Code

```
Public Shared ReadOnly SandyBrown As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.SeaGreen Field

A solid brush with the SeaGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush SeaGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly SeaGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.SeaShell Field

A solid brush with the SeaShell color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush SeaShell
```

Visual Basic Copy Code

```
Public Shared ReadOnly SeaShell As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Sienna Field

A solid brush with the Sienna color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Sienna
```

Visual Basic Copy Code

```
Public Shared ReadOnly Sienna As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Silver Field

A solid brush with the Silver color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Silver
```

Visual Basic Copy Code

```
Public Shared ReadOnly Silver As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.SkyBlue Field

A solid brush with the SkyBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush SkyBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly SkyBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.SlateBlue Field

A solid brush with the SlateBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush SlateBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly SlateBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.SlateGray Field

A solid brush with the SlateGray color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush SlateGray
```

Visual Basic Copy Code

```
Public Shared ReadOnly SlateGray As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Snow Field

A solid brush with the Snow color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Snow
```

Visual Basic Copy Code

```
Public Shared ReadOnly Snow As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.SpringGreen Field

A solid brush with the SpringGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush SpringGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly SpringGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.SteelBlue Field

A solid brush with the SteelBlue color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush SteelBlue
```

Visual Basic Copy Code

```
Public Shared ReadOnly SteelBlue As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Tan Field

A solid brush with the Tan color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Tan
```

Visual Basic Copy Code

```
Public Shared ReadOnly Tan As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Teal Field

A solid brush with the Teal color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Teal
```

Visual Basic Copy Code

```
Public Shared ReadOnly Teal As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Thistle Field

A solid brush with the Thistle color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Thistle
```

Visual Basic Copy Code

```
Public Shared ReadOnly Thistle As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Tomato Field

A solid brush with the Tomato color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Tomato
```

Visual Basic Copy Code

```
Public Shared ReadOnly Tomato As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Transparent Field

A solid brush with the Transparent color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Transparent
```

Visual Basic Copy Code

```
Public Shared ReadOnly Transparent As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Turquoise Field

A solid brush with the Turquoise color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Turquoise
```

Visual Basic Copy Code

```
Public Shared ReadOnly Turquoise As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Violet Field

A solid brush with the Violet color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Violet
```

Visual Basic Copy Code

```
Public Shared ReadOnly Violet As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Wheat Field

A solid brush with the Wheat color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Wheat
```

Visual Basic Copy Code

```
Public Shared ReadOnly Wheat As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.White Field

A solid brush with the White color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush White
```

Visual Basic Copy Code

```
Public Shared ReadOnly White As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.WhiteSmoke Field

A solid brush with the WhiteSmoke color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush WhiteSmoke
```

Visual Basic Copy Code

```
Public Shared ReadOnly WhiteSmoke As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.Yellow Field

A solid brush with the Yellow color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush Yellow
```

Visual Basic Copy Code

```
Public Shared ReadOnly Yellow As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

Brushes.YellowGreen Field

A solid brush with the YellowGreen color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly Brush YellowGreen
```

Visual Basic Copy Code

```
Public Shared ReadOnly YellowGreen As Brush
```

See Also

Brushes Members
Brushes Class
MindFusion.Drawing Namespace

CardinalSpline Class

Syntax

C# Copy Code

```
public class CardinalSpline
```

Visual Basic Copy Code

```
Public Class CardinalSpline
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.CardinalSpline

See Also

CardinalSpline Members
MindFusion.Drawing Namespace

CardinalSpline Members

The following tables list the members exposed by the CardinalSpline type.

Public Constructors

	Name	Description
	CardinalSpline	Overloaded.

Public Methods

	Name	Description
	AddToPath	

	Draw	
	GetPoints	

See Also

CardinalSpline Class
MindFusion.Drawing Namespace

CardinalSpline Constructor

Overload List

Name	Description
CardinalSpline (List(Point))	
CardinalSpline (List(Point), Nullable(Point), Nullable(Point))	

See Also

CardinalSpline Class
MindFusion.Drawing Namespace

CardinalSpline Constructor (List(Point))

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public CardinalSpline (
    List<Point> points
)
```

Visual Basic Copy Code

```
Public New ( _
    points As List(Of Point) _
)
```

Parameters

points

See Also

CardinalSpline Constructor Overload List
CardinalSpline Class
MindFusion.Drawing Namespace

CardinalSpline Constructor (List(Point), Nullable(Point), Nullable(Point))

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public CardinalSpline (
    List<Point> points,
    Nullable<Point> from,
    Nullable<Point> to
)
```

Visual Basic Copy Code

```
Public New ( _
    points As List(Of Point), _
    from As Nullable(Of Point), _
    to As Nullable(Of Point) _
)
```

Parameters

points

from

to

See Also

CardinalSpline Constructor Overload List
CardinalSpline Class
MindFusion.Drawing Namespace

CardinalSpline Methods

CardinalSpline.AddToPath Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void AddToPath (
    IGraphicsPath path
)
```

Visual Basic Copy Code

```
Public Sub AddToPath( _
    path As IGraphicsPath _
)
```

Parameters

path

See Also

CardinalSpline Members
CardinalSpline Class
MindFusion.Drawing Namespace

CardinalSpline.Draw Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void Draw (
    IGraphics graphics,
    Pen pen
)
```

Visual Basic Copy Code

```
Public Sub Draw( _
    graphics As IGraphics, _
    pen As Pen _
)
```

Parameters

graphics

pen

See Also

CardinalSpline Members
CardinalSpline Class
MindFusion.Drawing Namespace

CardinalSpline.GetPoints Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Point[] GetPoints ()
```

Visual Basic Copy Code

```
Public Function GetPoints () As Point()
```

Return Value

See Also

CardinalSpline Members
CardinalSpline Class
MindFusion.Drawing Namespace

CombineMode Enumeration

Specifies how different clipping regions can be combined.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum CombineMode
```

Visual Basic Copy Code

```
Public Enum CombineMode
```

Members

	Member name	Description
	Replace	One clipping region is replaced by another.
	Intersect	Two clipping regions are combined by taking their intersection.
	Union	Two clipping regions are combined by taking the union of both.
	Xor	Two clipping regions are combined by taking only the areas enclosed by one or the other region, but not both.
	Exclude	Specifies that the existing region is replaced by the result of the new region being removed from the existing region. Said differently, the new region is excluded from the existing region.
	Complement	Specifies that the existing region is replaced by the result of the existing region being removed from the new region. Said differently, the existing region is excluded from the new region.

See Also

MindFusion.Drawing Namespace

ContentAlignment Enumeration

Specifies alignment of content on the drawing surface.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum ContentAlignment
```

Visual Basic Copy Code

```
Public Enum ContentAlignment
```

Members

	Member name	Description
	BottomCenter	Content is vertically aligned at the bottom, and horizontally aligned at the center.
	BottomLeft	Content is vertically aligned at the bottom, and horizontally aligned on the left.
	BottomRight	Content is vertically aligned at the bottom, and horizontally aligned on the right.
	MiddleCenter	Content is vertically aligned in the middle, and horizontally aligned at the center.

	MiddleLeft	Content is vertically aligned in the middle, and horizontally aligned on the left.
	MiddleRight	Content is vertically aligned in the middle, and horizontally aligned on the right.
	TopCenter	Content is vertically aligned at the top, and horizontally aligned at the center.
	TopLeft	Content is vertically aligned at the top, and horizontally aligned on the left.
	TopRight	Content is vertically aligned at the top, and horizontally aligned on the right.

See Also

MindFusion.Drawing Namespace

CursorHint Enumeration

Specifies what mouse cursor to display while a user interacts with the diagram.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum CursorHint
```

Visual Basic Copy Code

```
Public Enum CursorHint
```

Members

	Member name	Description
	Move	Indicates the cursor specified by the MoveCursor property.
	Rotate	Indicates the cursor specified by the RotateCursor property.
	HorizontalResize	Indicates the cursor specified by the HorizontalResizeCursor property.
	VerticalResize	Indicates the cursor specified by the VerticalResizeCursor property.
	DiagonalResize	Indicates the cursor specified by the DiagonalResizeCursor property.
	CounterDiagonalResize	Indicates the cursor specified by the CounterDiagonalResizeCursor property.
	DrawLink	Indicates the cursor specified by the DrawLinkCursor property.
	Pointer	Indicates the cursor specified by the PointerCursor property.
	AllowLink	Indicates the cursor specified by the AllowLinkCursor property.
	DisallowLink	Indicates the cursor specified by the DisallowLinkCursor property.
	Disallow	Indicates the cursor specified by the DisallowCursor property.
	DontChange	Indicates the cursor specified by the Cursor property.
	EditText	Indicates the cursor specified by the EditTextCursor property.

See Also

DashStyle Enumeration

Specifies the style of dashed lines.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum DashStyle
```

Visual Basic Copy Code

```
Public Enum DashStyle
```

Members

	Member name	Description
	Solid	Specifies a solid line.
	Dash	Specifies a line consisting of dashes.
	Dot	Specifies a line consisting of dots.
	DashDot	Specifies a line consisting of a repeating pattern of dash-dot.
	DashDotDot	Specifies a line consisting of a repeating pattern of dash-dot-dot.
	Custom	Specifies a user-defined dash style.

See Also

MindFusion.Drawing Namespace

FillMode Enumeration

Specifies how the interior of a closed path is filled.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum FillMode
```

Visual Basic Copy Code

```
Public Enum FillMode
```

Members

	Member name	Description
	Alternate	Specifies the alternate fill mode.
	Winding	Specifies the winding fill mode.

See Also

MindFusion.Drawing Namespace

Font Class

Encapsulates font family, size and style information.

Syntax

C# Copy Code

```
public class Font : ICloneable
```

Visual Basic Copy Code

```
Public Class Font
    Implements ICloneable
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.Font

See Also

Font Members
MindFusion.Drawing Namespace

Font Members

Encapsulates font family, size and style information.

The following tables list the members exposed by the Font type.

Public Constructors

	Name	Description
	Font	Overloaded.

Public Properties

	Name	Description
	Bold	Gets whether the font is bold.
	Height	Gets the font height.
	Italic	Gets whether the font is italic.
	Name	Gets or sets the font name.
	Size	Gets or sets the font size.
	SizeInPoints	Gets the font height.
	Strikeout	Gets whether the font draws a line through the text.
	Style	Gets or sets the font style.
	Underline	Gets whether the font is underlined.
	Unit	Gets or sets the unit used to specify font size.

Public Methods

	Name	Description
	Clone	Creates a copy of this Font.

See Also

Font Class
MindFusion.Drawing Namespace

Font Constructor

Overload List

Name	Description
Font (String, Double)	Initializes a new instance of the Font class.
Font (String, Double, FontStyle)	Initializes a new instance of the Font class.
Font (String, Double, FontStyle, GraphicsUnit)	Initializes a new instance of the Font class.

See Also

Font Class
MindFusion.Drawing Namespace

Font Constructor (String, Double)

Initializes a new instance of the Font class.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Font (
    string name,
    double size
)
```

Visual Basic Copy Code

```
Public New ( _
    name As String, _
    size As Double _
)
```

Parameters

name

size

See Also

Font Constructor Overload List
Font Class
MindFusion.Drawing Namespace

Font Constructor (String, Double, FontStyle)

Initializes a new instance of the Font class.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Font (
    string name,
    double size,
    FontStyle style
)
```

Visual Basic Copy Code

```
Public New ( _
    name As String, _
    size As Double, _
    style As FontStyle _
)
```

Parameters

name

size

style

See Also

Font Constructor Overload List
Font Class
MindFusion.Drawing Namespace

Font Constructor (String, Double, FontStyle, GraphicsUnit)

Initializes a new instance of the Font class.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Font (
    string name,
    double size,
    FontStyle style,
    GraphicsUnit unit
)
```

Visual Basic Copy Code

```
Public New ( _
    name As String, _
    size As Double, _
    style As FontStyle, _
    unit As GraphicsUnit _
)
```

Parameters

name

size

style

unit

See Also

Font Constructor Overload List
Font Class
MindFusion.Drawing Namespace

Font Methods

Font.Clone Method

Creates a copy of this Font.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Object Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As Object
```

Return Value

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font Properties

Font.Bold Property

Gets whether the font is bold.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public bool Bold { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Bold As Boolean
```

Property Value

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.Height Property

Gets the font height.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public int Height { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Height As Integer
```

Property Value

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.Italic Property

Gets whether the font is italic.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public bool Italic { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Italic As Boolean
```

Property Value

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.Name Property

Gets or sets the font name.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
public string Name { get; set; }
```

Visual Basic

```
Public Property Name As String
```

Property Value

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.Size Property

Gets or sets the font size.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
public double Size { get; set; }
```

Visual Basic

```
Public Property Size As Double
```

Property Value

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.SizeInPoints Property

Gets the font height.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
public double SizeInPoints { get; }
```

Visual Basic

```
Public ReadOnly Property SizeInPoints As Double
```

Property Value

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.Strikeout Property

Gets whether the font draws a line through the text.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public bool Strikeout { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Strikeout As Boolean
```

Property Value

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.Style Property

Gets or sets the font style.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public FontStyle Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As FontStyle
```

Property Value

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.Underline Property

Gets whether the font is underlined.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public bool Underline { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Underline As Boolean
```

Property Value

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

Font.Unit Property

Gets or sets the unit used to specify font size.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public GraphicsUnit Unit { get; set; }
```

Visual Basic Copy Code

```
Public Property Unit As GraphicsUnit
```

Property Value

See Also

Font Members
Font Class
MindFusion.Drawing Namespace

FontManager Class

Maps font names to font files.

Syntax

C# Copy Code

```
public sealed class FontManager
```

Visual Basic Copy Code

```
Public NotInheritable Class FontManager
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.FontManager

See Also

FontManager Members

Maps font names to font files.

The following tables list the members exposed by the FontManager type.

Public Methods

	Name	Description
	Register	Overloaded.

See Also

FontManager Class
MindFusion.Drawing Namespace

FontManager Methods

FontManager.Register Method

Overload List

Name	Description
FontManager.Register (Stream)	Registers a font from specified font-file stream.
FontManager.Register (Stream, String)	Registers a font from specified font-file stream.

See Also

FontManager Members
FontManager Class
MindFusion.Drawing Namespace

FontManager.Register Method (Stream)

Registers a font from specified font-file stream.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static void Register (
    Stream fontFile
)
```

Visual Basic Copy Code

```
Public Shared Sub Register( _
    fontFile As Stream _
)
```

Parameters

fontFile
A stream containing font file.

See Also

Register Method Overload List
FontManager Members
FontManager Class
MindFusion.Drawing Namespace

FontManager.Register Method (Stream, String)

Registers a font from specified font-file stream.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static void Register (
    Stream fontFile,
    string overrideName
)
```

Visual Basic Copy Code

```
Public Shared Sub Register( _
    fontFile As Stream, _
    overrideName As String _
)
```

Parameters

fontFile
A stream containing font file.

overrideName
The name to use for specified font.

See Also

Register Method Overload List
FontManager Members
FontManager Class
MindFusion.Drawing Namespace

FontStyle Enumeration

Specifies font style.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum FontStyle
```

Visual Basic Copy Code

```
Public Enum FontStyle
```

Members

	Member name	Description
	Regular	Normal text.
	Bold	Bold text.
	Italic	Italic text.
	Underline	Underlined text.
	Strikeout	Text with a line through the middle.

See Also

MindFusion.Drawing Namespace

GraphicsFactory Class

A factory class for various graphical objects.

Syntax

C# Copy Code

```
public sealed class GraphicsFactory
```

Visual Basic Copy Code

```
Public NotInheritable Class GraphicsFactory
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.GraphicsFactory

See Also

GraphicsFactory Members
MindFusion.Drawing Namespace

GraphicsFactory Members

A factory class for various graphical objects.

The following tables list the members exposed by the GraphicsFactory type.

Public Methods

	Name	Description
	CreateImageGraphics	Overloaded.
	CreateMatrix	Overloaded.
	CreatePath	Overloaded.

See Also

GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsFactory Methods

GraphicsFactory.CreateImageGraphics Method

Overload List

Name	Description
GraphicsFactory.CreateImageGraphics ()	Creates an IGraphics object specifically to output to an image.
GraphicsFactory.CreateImageGraphics (Rect)	Creates an IGraphics object specifically to output to an image.
GraphicsFactory.CreateImageGraphics (Rect, SKBitmap@)	Creates an IGraphics object specifically to output to an image.

See Also

GraphicsFactory Members
GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsFactory.CreateImageGraphics Method ()

Creates an IGraphics object specifically to output to an image.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static IGraphics CreateImageGraphics ()
```

Visual Basic Copy Code

```
Public Shared Function CreateImageGraphics () As IGraphics
```

Return Value

An instance of the IGraphics interface.

See Also

CreateImageGraphics Method Overload List
GraphicsFactory Members
GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsFactory.CreateImageGraphics Method (Rect)

Creates an IGraphics object specifically to output to an image.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static IGraphics CreateImageGraphics (
    Rect bounds
)
```

Visual Basic

```
Public Shared Function CreateImageGraphics( _
    bounds As Rect _
) As IGraphics
```

Parameters

bounds

Return Value

An instance of the IGraphics interface.

See Also

CreateImageGraphics Method Overload List
GraphicsFactory Members
GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsFactory.CreateImageGraphics Method (Rect, SKBitmap@)

Creates an IGraphics object specifically to output to an image.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
public static IGraphics CreateImageGraphics (
    Rect bounds,
    out SKBitmap bitmap
)
```

Visual Basic

```
Public Shared Function CreateImageGraphics( _
    bounds As Rect, _
    <System.Runtime.InteropServices.Out()> ByRef bitmap As SKBitmap _
) As IGraphics
```

Parameters

bounds

bitmap

Return Value

An instance of the IGraphics interface.

See Also

CreateImageGraphics Method Overload List
GraphicsFactory Members
GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsFactory.CreateMatrix Method

Overload List

Name	Description
GraphicsFactory.CreateMatrix ()	Creates an IMatrix object.

GraphicsFactory.CreateMatrix (Single, Single, Single, Single, Single, Single)	Creates an IMatrix object.

See Also

GraphicsFactory Members
GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsFactory.CreateMatrix Method ()

Creates an IMatrix object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static IMatrix CreateMatrix ()
```

Visual Basic Copy Code

```
Public Shared Function CreateMatrix () As IMatrix
```

Return Value

An instance of the IMatrix interface.

See Also

CreateMatrix Method Overload List
GraphicsFactory Members
GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsFactory.CreateMatrix Method (Single, Single, Single, Single, Single, Single)

Creates an IMatrix object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static IMatrix CreateMatrix (
    float m11,
    float m12,
    float m21,
    float m22,
    float m31,
    float m32
)
```

Visual Basic Copy Code

```
Public Shared Function CreateMatrix( _
    m11 As Single, _
    m12 As Single, _
    m21 As Single, _
    m22 As Single, _
    m31 As Single, _
```

```
    m32 As Single _
) As IMatrix
```

Parameters

m11

m12

m21

m22

m31

m32

Return Value

An instance of the IMatrix interface.

See Also

CreateMatrix Method Overload List
GraphicsFactory Members
GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsFactory.CreatePath Method

Overload List

Name	Description
GraphicsFactory.CreatePath ()	Creates an IGraphicsPath object.
GraphicsFactory.CreatePath (FillMode)	Creates an IGraphicsPath object with the specified fill mode.
GraphicsFactory.CreatePath (Rect)	Creates an IGraphicsPath object from the specified rectangle.

See Also

GraphicsFactory Members
GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsFactory.CreatePath Method ()

Creates an IGraphicsPath object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static IGraphicsPath CreatePath ()
```

Visual Basic Copy Code

```
Public Shared Function CreatePath () As IGraphicsPath
```

Return Value

An instance of the IGraphicsPath interface.

See Also

CreatePath Method Overload List
GraphicsFactory Members
GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsFactory.CreatePath Method (FillMode)

Creates an IGraphicsPath object with the specified fill mode.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static IGraphicsPath CreatePath (
    FillMode fillMode
)
```

Visual Basic Copy Code

```
Public Shared Function CreatePath( _
    fillMode As FillMode _
) As IGraphicsPath
```

Parameters

fillMode
A member of the FillMode enumeration.

Return Value

An instance of the IGraphicsPath interface.

See Also

CreatePath Method Overload List
GraphicsFactory Members
GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsFactory.CreatePath Method (Rect)

Creates an IGraphicsPath object from the specified rectangle.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static IGraphicsPath CreatePath (
    Rect rect
)
```

Visual Basic Copy Code

```
Public Shared Function CreatePath( _
    rect As Rect _
) As IGraphicsPath
```

Parameters

rect
A Rectangle structure.

Return Value

An instance of the IGraphicsPath interface.

See Also

CreatePath Method Overload List
GraphicsFactory Members
GraphicsFactory Class
MindFusion.Drawing Namespace

GraphicsType Enumeration

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum GraphicsType
```

Visual Basic Copy Code

```
Public Enum GraphicsType
```

Members

	Member name	Description
	Generic	
	PDF	

See Also

MindFusion.Drawing Namespace

GraphicsUnit Enumeration

Specifies unit of measure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum GraphicsUnit
```

Visual Basic Copy Code

```
Public Enum GraphicsUnit
```

Members

	Member name	Description
	World	Match the measure unit of parent object.
	Display	Specifies the unit of measure of the display device.
	Pixel	Specifies a device pixel as the unit of measure.

	Point	A point (1/72 inch) as the unit of measure.
	Inch	Specifies the inch as the unit of measure.
	Document	Document unit (1/300 inch) as the unit of measure.
	Millimeter	Specifies the millimeter as the unit of measure.
	WpfPoint	WPF point (1/96 inch) as the unit of measure.
	Percent	Specifies percents as the unit of measure.
	Centimeter	Specifies the centimeter as the unit of measure.

See Also

MindFusion.Drawing Namespace

HatchBrush Class

HatchBrush fills shapes using a hatch style, a foreground color, and a background color.

Syntax

C# Copy Code

```
public class HatchBrush : Brush
```

Visual Basic Copy Code

```
Public Class HatchBrush
    Inherits Brush
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.Brush
    MindFusion.Drawing.HatchBrush

See Also

HatchBrush Members
MindFusion.Drawing Namespace

HatchBrush Members

HatchBrush fills shapes using a hatch style, a foreground color, and a background color.

The following tables list the members exposed by the **HatchBrush** type.

Public Constructors

	Name	Description
	HatchBrush	Initializes a new instance of the **HatchBrush** class with the specified style, foreground color and background color.

Public Properties

	Name	Description
	BackgroundColor	Gets or sets the color of spaces between the hatch lines drawn by this brush.
	ForegroundColor	Gets or sets the color of hatch lines drawn by this brush.

	Style	Gets the hatch style of this brush.

Public Methods

	Name	Description
	Clone	Creates an exact copy of this brush.
	Equals	Brush.Equals override.
	GetHashCode	Brush.GetHashCode override.
	ToColor	Brush.ToColor override.

See Also

HatchBrush Class
MindFusion.Drawing Namespace

HatchBrush Constructor

Initializes a new instance of the HatchBrush class with the specified style, foreground color and background color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public HatchBrush (
    HatchStyle style,
    Color foregroundColor,
    Color backgroundColor
)
```

Visual Basic Copy Code

```
Public New ( _
    style As HatchStyle, _
    foregroundColor As Color, _
    backgroundColor As Color _
)
```

Parameters

style

foregroundColor

backgroundColor

See Also

HatchBrush Class
MindFusion.Drawing Namespace

HatchBrush Methods

HatchBrush.Clone Method

Creates an exact copy of this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override Brush Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Brush
```

Return Value

An instance of the HatchBrush class.

See Also

HatchBrush Members
HatchBrush Class
MindFusion.Drawing Namespace

HatchBrush.Equals Method

Brush.Equals override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override bool Equals (
    Brush other
)
```

Visual Basic Copy Code

```
Public Overrides Function Equals( _
    other As Brush _
) As Boolean
```

Parameters

other
The Brush to compare with the current Brush.

Return Value

true if the specified Brush is equal to the current Brush; otherwise, false.

See Also

HatchBrush Members
HatchBrush Class
MindFusion.Drawing Namespace

HatchBrush.GetHashCode Method

Brush.GetHashCode override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override int GetHashCode ()
```

Visual Basic Copy Code

```
Public Overrides Function GetHashCode () As Integer
```

Return Value

See Also

HatchBrush Members
HatchBrush Class
MindFusion.Drawing Namespace

HatchBrush.ToColor Method

Brush.ToColor override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override Color ToColor ()
```

Visual Basic Copy Code

```
Public Overrides Function ToColor () As Color
```

Return Value

See Also

HatchBrush Members
HatchBrush Class
MindFusion.Drawing Namespace

HatchBrush Properties

HatchBrush.BackgroundColor Property

Gets or sets the color of spaces between the hatch lines drawn by this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Color BackgroundColor { get; set; }
```

Visual Basic Copy Code

```
Public Property BackgroundColor As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** that represents the background color of this brush.

See Also

HatchBrush Members
HatchBrush Class
MindFusion.Drawing Namespace

HatchBrush.ForegroundColor Property

Gets or sets the color of hatch lines drawn by this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Color ForegroundColor { get; set; }
```

Visual Basic Copy Code

```
Public Property ForegroundColor As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** that represents the foreground color of this brush.

See Also

HatchBrush Members
HatchBrush Class
MindFusion.Drawing Namespace

HatchBrush.Style Property

Gets the hatch style of this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public HatchStyle Style { get; set; }
```

Visual Basic Copy Code

```
Public Property Style As HatchStyle
```

Property Value

See Also

HatchBrush Members
HatchBrush Class
MindFusion.Drawing Namespace

HatchStyle Enumeration

Represents the available hatch patterns.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum HatchStyle
```

Visual Basic Copy Code

```
Public Enum HatchStyle
```

Members

	Member name	Description
	Min	Specifies hatch style Horizontal.
	Horizontal	A pattern of horizontal lines.
	Vertical	A pattern of vertical lines.
	ForwardDiagonal	A pattern of lines on a diagonal from upper left to lower right.
	BackwardDiagonal	A pattern of lines on a diagonal from upper right to lower left.
	Cross	Specifies horizontal and vertical lines that cross.
	Max	Specifies hatch style SolidDiamond.
	LargeGrid	Specifies the hatch style Cross.
	DiagonalCross	A pattern of crisscross diagonal lines.
	Percent05	Specifies a 5-percent hatch. The ratio of foreground color to background color is 5:100.
	Percent10	Specifies a 10-percent hatch. The ratio of foreground color to background color is 10:100.
	Percent20	Specifies a 20-percent hatch. The ratio of foreground color to background color is 20:100.
	Percent25	Specifies a 25-percent hatch. The ratio of foreground color to background color is 25:100.
	Percent30	Specifies a 30-percent hatch. The ratio of foreground color to background color is 30:100.
	Percent40	Specifies a 40-percent hatch. The ratio of foreground color to background color is 40:100.
	Percent50	Specifies a 50-percent hatch. The ratio of foreground color to background color is 50:100.
	Percent60	Specifies a 60-percent hatch. The ratio of foreground color to background color is 60:100.
	Percent70	Specifies a 70-percent hatch. The ratio of foreground color to background color is 70:100.
	Percent75	Specifies a 75-percent hatch. The ratio of foreground color to background color is 75:100.
	Percent80	Specifies a 80-percent hatch. The ratio of foreground color to background color is 80:100.
	Percent90	Specifies a 90-percent hatch. The ratio of foreground color to background color is 90:100.
	LightDownwardDiagonal	Specifies diagonal lines that slant to the right from top points to bottom points and are spaced 50 percent closer together than ForwardDiagonal, but are not antialiased.

LightUpwardDiagonal	Specifies diagonal lines that slant to the left from top points to bottom points and are spaced 50 percent closer together than BackwardDiagonal.
DarkDownwardDiagonal	Specifies diagonal lines that slant to the right from top points to bottom points, are spaced 50 percent closer together than, and are twice the width of ForwardDiagonal.
DarkUpwardDiagonal	Specifies diagonal lines that slant to the left from top points to bottom points, are spaced 50 percent closer together than BackwardDiagonal, and are twice its width.
WideDownwardDiagonal	Specifies diagonal lines that slant to the right from top points to bottom points, have the same spacing as hatch style ForwardDiagonal, and are triple its width.
WideUpwardDiagonal	Specifies diagonal lines that slant to the left from top points to bottom points, have the same spacing as hatch style BackwardDiagonal, and are triple its width.
LightVertical	Specifies vertical lines that are spaced 50 percent closer together than Vertical.
LightHorizontal	Specifies horizontal lines that are spaced 50 percent closer together than Horizontal.
NarrowVertical	Specifies vertical lines that are spaced 75 percent closer together than hatch style Vertical.
NarrowHorizontal	Specifies horizontal lines that are spaced 75 percent closer together than hatch style Horizontal.
DarkVertical	Specifies vertical lines that are spaced 50 percent closer together than Vertical and are twice its width.
DarkHorizontal	Specifies horizontal lines that are spaced 50 percent closer together than Horizontal and are twice the width of Horizontal.
DashedDownwardDiagonal	Specifies dashed diagonal lines, that slant to the right from top points to bottom points.
DashedUpwardDiagonal	Specifies dashed diagonal lines, that slant to the left from top points to bottom points.
DashedHorizontal	Specifies dashed horizontal lines.
DashedVertical	Specifies dashed vertical lines.
SmallConfetti	Specifies a hatch that has the appearance of confetti.
LargeConfetti	Specifies a hatch that has the appearance of confetti, and is composed of larger pieces than SmallConfetti.
ZigZag	Specifies horizontal lines that are composed of zigzags.
Wave	Specifies horizontal lines that are composed of tildes.
DiagonalBrick	Specifies a hatch that has the appearance of layered bricks that slant to the left from top points to bottom points.
HorizontalBrick	Specifies a hatch that has the appearance of horizontally layered bricks.
Weave	Specifies a hatch that has the appearance of a woven material.
Plaid	Specifies a hatch that has the appearance of a plaid material.
Divot	Specifies a hatch that has the appearance of divots.
DottedGrid	Specifies horizontal and vertical lines, each of which is composed of dots, that cross.
DottedDiamond	Specifies forward diagonal and backward diagonal lines, each of which is composed of dots, that cross.

	Shingle	Specifies a hatch that has the appearance of diagonally layered shingles that slant to the right from top points to bottom points.
	Trellis	Specifies a hatch that has the appearance of a trellis.
	Sphere	Specifies a hatch that has the appearance of spheres laid adjacent to one another.
	SmallGrid	Specifies horizontal and vertical lines that cross and are spaced 50 percent closer together than hatch style Cross.
	SmallCheckerBoard	Specifies a hatch that has the appearance of a checkerboard.
	LargeCheckerBoard	Specifies a hatch that has the appearance of a checkerboard with squares that are twice the size of SmallCheckerBoard.
	OutlinedDiamond	Specifies forward diagonal and backward diagonal lines that cross.
	SolidDiamond	Specifies a hatch that has the appearance of a checkerboard placed diagonally.

See Also

MindFusion.Drawing Namespace

HorizontalAlignment Enumeration

Specifies the horizontal alignment of an element.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum HorizontalAlignment
```

Visual Basic Copy Code

```
Public Enum HorizontalAlignment
```

Members

	Member name	Description
	Left	Indicates that an element is left-aligned to its container.
	Center	Indicates that an element is centered in its container.
	Right	Indicates that an element is right-aligned in its container.

See Also

MindFusion.Drawing Namespace

IGraphics Interface

This interface exposes methods for drawing graphics primitives used to render MindFusion.Diagramming objects.

Syntax

C# Copy Code

```
public interface IGraphics : IDisposable
```

Visual Basic Copy Code

```
Public Interface IGraphics
    Implements IDisposable
```

Inheritors

MindFusion.Drawing.MauiGraphics

See Also

IGraphics Members
MindFusion.Drawing Namespace

IGraphics Members

This interface exposes methods for drawing graphics primitives used to render MindFusion.Diagramming objects.

The following tables list the members exposed by the IGraphics type.

Public Properties

	Name	Description
	Clip	Gets or sets an IGraphicsPath object that limits the drawing region of this **IGraphics** object.
	DpiX	Gets the horizontal resolution of this **IGraphics** object.
	DpiY	Gets the vertical resolution of this **IGraphics** object.
	EnableAntialiasing	Gets or sets whether to anti-alias the graphics primitives.
	SupportsPenWidth0	Gets a value indicating whether the IGraphics object supports pens with zero width.
	Transform	Gets or sets the world transformation for this **IGraphics** object.
	UnitScale	Gets the scale for current unit.

Public Methods

	Name	Description
	BeginHyperLink	
	ClipImage	Overloaded.
	CreateImage	Creates an image from the current drawing associated with this IGraphics object.
	DrawArc	Overloaded. Draws an arc representing a portion of an ellipse specified by a pair of coordinates, a width, and a height.
	DrawBezier	Draws a Bézier spline defined by four **Point** structures.
	DrawBeziers	Draws a series of Bézier splines from an array of Point structures.
	DrawCurve	Draws a cardinal spline through a specified array of **Point** structures.
	DrawEllipse	Overloaded. Draws an ellipse defined by a bounding rectangle specified by a pair of coordinates, a height, and a width.
	DrawImage	Overloaded. Draws the specified **IImage** object at the specified location and with the original size.
	DrawLine	Overloaded. Draws a line segment connecting two points.

	DrawLines	Draws a series of line segments that connect an array of Point structures.
	DrawPath	Draws an **IGraphicsPath** object.
	DrawPoint	Draws a point at the specified coordinates.
	DrawPolygon	Draws a polygon specified by a poins list.
	DrawRectangle	Overloaded. Draws a rectangle specified by a coordinate pair, a width, and a height.
	DrawString	Overloaded. Draws the specified text string at the specified location with the specified **Brush** and **Font** objects.
	EndHyperLink	
	FillEllipse	Overloaded. Fills the interior of an ellipse defined by a bounding rectangle specified by a pair of coordinates, a width, and a height.
	FillPath	Fills the interior of a **GraphicsPath** object.
	FillPolygon	Fills the interior of a polygon specified by a points list.
	FillRectangle	Overloaded. Fills the interior of a rectangle specified by a pair of coordinates, a width, and a height.
	GetFontHeight	Gets the height of the specified font.
	GetImageSize	Returns the physical size of the specified image.
	MeasureString	Overloaded. Measures the specified string when drawn with the specified **Font** object.
	MultiplyTransform	Multiplies the world transformation of this **IGraphics** object and specified the **Matrix** object in the specified order.
	PopOpacity	Restores global opacity.
	PushOpacity	Sets global opacity.
	ResetTransform	Resets the world transformation matrix of this **IGraphics** object to the identity matrix.
	Restore	Restores the state of this **IGraphics** object to the specified state argument.
	RotateTransform	Applies the specified rotation to the transformation matrix of this **IGraphics** object.
	Save	Saves the current state of this **IGraphics** object.
	ScaleTransform	Applies the specified scaling operation to the transformation matrix of this **IGraphics** object by prepending it to the object's transformation matrix.
	SetClip	Overloaded. Sets the clipping region of this **IGraphics** object to the **Clip** property of the specified **IGraphics** object.
	SetTransform	Sets the current transformation.
	TranslateTransform	Prepends the specified translation to the transformation matrix of this **IGraphics** object.

See Also

IGraphics Interface
MindFusion.Drawing Namespace

IGraphics Methods

IGraphics.BeginHyperLink Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void BeginHyperLink (
    string hyperlink,
    string target
)
```

Visual Basic Copy Code

```
Sub BeginHyperLink( _
    hyperlink As String, _
    target As String _
)
```

Parameters

hyperlink

target

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.ClipImage Method

Overload List

Name	Description
IGraphics.ClipImage (IImage, IGraphicsPath)	Applies a clipping path to an image.
IGraphics.ClipImage (IImage, Point[])	Applies a clipping path to an image.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.ClipImage Method (IImage, IGraphicsPath)

Applies a clipping path to an image.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
IImage ClipImage (
    IImage sourceImage,
    IGraphicsPath clip
)
```

Visual Basic

```
Function ClipImage( _
    sourceImage As IImage, _
    clip As IGraphicsPath _
) As IImage
```

Parameters

sourceImage
The image to clip.

clip
The path used for the clip.

Return Value

The clipped image.

See Also

ClipImage Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.ClipImage Method (IImage, Point[])

Applies a clipping path to an image.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
IImage ClipImage (
    IImage sourceImage,
    Point[] clipPoly
)
```

Visual Basic

```
Function ClipImage( _
    sourceImage As IImage, _
    clipPoly() As Point _
) As IImage
```

Parameters

sourceImage
The image to clip.

clipPoly
A polygon defined by a point collection used for the clip.

Return Value

The clipped image.

See Also

ClipImage Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.CreateImage Method

Creates an image from the current drawing associated with this IGraphics object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
IImage CreateImage (
    IGraphicsPath clip = null
)
```

Visual Basic Copy Code

```
Function CreateImage( _
    Optional clip As IGraphicsPath = Nothing _
) As IImage
```

Parameters

clip
Optional.
An optional clip path.

Return Value

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawArc Method

Draws an arc representing a portion of an ellipse specified by a pair of coordinates, a width, and a height.

Overload List

Name	Description
IGraphics.DrawArc (Pen, Double, Double, Double, Double, Double, Double)	Draws an arc representing a portion of an ellipse specified by a pair of coordinates, a width, and a height.
IGraphics.DrawArc (Pen, Rect, Double, Double)	Draws an arc representing a portion of an ellipse specified by a **Rect** structure.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawArc Method (Pen, Double, Double, Double, Double, Double, Double)

Draws an arc representing a portion of an ellipse specified by a pair of coordinates, a width, and a height.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawArc (
    Pen pen,
    double x,
    double y,
    double width,
    double height,
    double startAngle,
    double sweepAngle
)
```

Visual Basic Copy Code

```
Sub DrawArc( _
    pen As Pen, _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double, _
    startAngle As Double, _
    sweepAngle As Double _
)
```

Parameters

pen
A Pen object that determines the color, width, and style of the arc.

x
The x-coordinate of the upper-left corner of the rectangle that defines the ellipse.

y
The y-coordinate of the upper-left corner of the rectangle that defines the ellipse.

width
The width of the rectangle that defines the ellipse.

height
The height of the rectangle that defines the ellipse.

startAngle
Angle in degrees measured clockwise from the x-axis to the starting point of the arc.

sweepAngle
Angle in degrees measured clockwise from the *startAngle* parameter to ending point of the arc.

See Also

DrawArc Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawArc Method (Pen, Rect, Double, Double)

Draws an arc representing a portion of an ellipse specified by a **Rect** structure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawArc (
    Pen pen,
    Rect rect,
    double startAngle,
    double sweepAngle
)
```

Visual Basic Copy Code

```
Sub DrawArc( _
    pen As Pen, _
    rect As Rect, _
    startAngle As Double, _
    sweepAngle As Double _
)
```

Parameters

pen

A Pen object that determines the color, width, and style of the arc.

rect

A **Microsoft.Maui.Graphics.Rect** that defines the boundaries of the ellipse.

startAngle

Angle in degrees measured clockwise from the x-axis to the starting point of the arc.

sweepAngle

Angle in degrees measured clockwise from the *startAngle* parameter to ending point of the arc.

See Also

DrawArc Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawBezier Method

Draws a Bézier spline defined by four **Point** structures.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawBezier (
    Pen pen,
    Point pt1,
    Point pt2,
    Point pt3,
    Point pt4
)
```

Visual Basic Copy Code

```
Sub DrawBezier( _
    pen As Pen, _
    pt1 As Point, _
    pt2 As Point, _
    pt3 As Point, _
    pt4 As Point _
)
```

Parameters

pen

A Pen object that determines the color, width, and style of the curve.

pt1

A **Microsoft.Maui.Graphics.Point** that represents the starting point of the curve.

pt2

A **Microsoft.Maui.Graphics.Point** that represents the first control point for the curve.

pt3

A **Microsoft.Maui.Graphics.Point** that represents the second control point for the curve.

pt4

A **Microsoft.Maui.Graphics.Point** that represents the ending point of the curve.

See Also

DrawBezier Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawBeziers Method

Draws a series of Bézier splines from an array of Point structures.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawBeziers (
    Pen pen,
    IList<Point> points
)
```

Visual Basic Copy Code

```
Sub DrawBeziers( _
    pen As Pen, _
    points As IList(Of Point) _
)
```

Parameters

pen
A Pen object that determines the color, width, and style of the curve.

points
An array of Point structures that represent the points that determine the curve.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawCurve Method

Draws a cardinal spline through a specified array of **Point** structures.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawCurve (
    Pen pen,
    Point[] points
)
```

Visual Basic Copy Code

```
Sub DrawCurve( _
    pen As Pen, _
    points() As Point _
)
```

Parameters

pen
A Pen object that determines the color, width, and height of the curve.

points
A **Microsoft.Maui.Graphics.Point** array that defines the spline.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawEllipse Method

Draws an ellipse defined by a bounding rectangle specified by a pair of coordinates, a height, and a width.

Overload List

Name	Description
IGraphics.DrawEllipse (Pen, Double, Double, Double, Double)	Draws an ellipse defined by a bounding rectangle specified by a pair of coordinates, a height, and a width.
IGraphics.DrawEllipse (Pen, Rect)	Draws an ellipse defined by a bounding rectangle.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawEllipse Method (Pen, Double, Double, Double, Double)

Draws an ellipse defined by a bounding rectangle specified by a pair of coordinates, a height, and a width.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawEllipse (
    Pen pen,
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Sub DrawEllipse( _
    pen As Pen, _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
)
```

Parameters

pen
A Pen object that determines the color, width, and style of the ellipse.

x
The x-coordinate of the upper-left corner of the bounding rectangle that defines the ellipse.

y
The y-coordinate of the upper-left corner of the bounding rectangle that defines the ellipse.

width
The width of the bounding rectangle that defines the ellipse.

height
The height of the bounding rectangle that defines the ellipse.

See Also

DrawEllipse Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawEllipse Method (Pen, Rect)

Draws an ellipse defined by a bounding rectangle.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawEllipse (
    Pen pen,
    Rect rect
)
```

Visual Basic Copy Code

```
Sub DrawEllipse( _
    pen As Pen, _
    rect As Rect _
)
```

Parameters

pen
A Pen object that determines the color, width, and style of the ellipse.

rect
A **Microsoft.Maui.Graphics.Rect** that defines the boundaries of the ellipse.

See Also

DrawEllipse Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawImage Method

Draws the specified **IImage** object at the specified location and with the original size.

Overload List

Name	Description
IGraphics.DrawImage (IImage, Double, Double)	Draws the specified image, using its original physical size, at the specified location.
IGraphics.DrawImage (IImage, Rect)	Draws the specified image at the specified location and with the specified size.

IGraphics.DrawImage (IImage, Rect, ImageAlign, Double, Point)	Draws the image represented by the specified IImage object at the specified location, with the specified alignment and rotation.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawImage Method (IImage, Double, Double)

Draws the specified image, using its original physical size, at the specified location.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawImage (
    IImage image,
    double x,
    double y
)
```

Visual Basic Copy Code

```
Sub DrawImage( _
    image As IImage, _
    x As Double, _
    y As Double _
)
```

Parameters

image
A **Microsoft.Maui.Graphics.IImage** to draw.

x
The x-coordinate of the upper-left corner of the drawn image.

y
The y-coordinate of the upper-left corner of the drawn image.

See Also

DrawImage Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawImage Method (IImage, Rect)

Draws the specified image at the specified location and with the specified size.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawImage (
    IImage image,
    Rect rect
)
```

Visual Basic

```
Sub DrawImage( _
    image As IImage, _
    rect As Rect _
)
```

Parameters

image

A **Microsoft.Maui.Graphics.IImage** to draw.

rect

A **Microsoft.Maui.Graphics.Rect** that specifies the location and size of the drawn image.

See Also

DrawImage Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawImage Method (IImage, Rect, ImageAlign, Double, Point)

Draws the image represented by the specified IImage object at the specified location, with the specified alignment and rotation.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
void DrawImage (
    IImage image,
    Rect rect,
    ImageAlign imageAlign,
    double rotationAngle,
    Point pivot
)
```

Visual Basic

```
Sub DrawImage( _
    image As IImage, _
    rect As Rect, _
    imageAlign As ImageAlign, _
    rotationAngle As Double, _
    pivot As Point _
)
```

Parameters

image

A **Microsoft.Maui.Graphics.IImage** representing the image to draw.

rect

A **Microsoft.Maui.Graphics.Rect** that specifies the location and size of the drawn image.

imageAlign

The alignment of the image within the specified rectangle.

rotationAngle

The rotation angle of the drawn image.

pivot

The pivot of the rotation.

See Also

DrawImage Method Overload List
IGraphics Members

IGraphics.DrawLine Method

Draws a line segment connecting two points.

Overload List

Name	Description
IGraphics.DrawLine (Pen, Double, Double, Double, Double)	Draws a line segment connecting the two points specified by coordinate pairs.
IGraphics.DrawLine (Pen, Point, Point)	Draws a line segment connecting two points.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawLine Method (Pen, Double, Double, Double, Double)

Draws a line segment connecting the two points specified by coordinate pairs.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawLine (
    Pen pen,
    double x1,
    double y1,
    double x2,
    double y2
)
```

Visual Basic Copy Code

```
Sub DrawLine( _
    pen As Pen, _
    x1 As Double, _
    y1 As Double, _
    x2 As Double, _
    y2 As Double _
)
```

Parameters

pen
A Pen object that determines the color, width, and style of the line.

x1
The x-coordinate of the first point.

y1
The y-coordinate of the first point.

x2
The x-coordinate of the second point.

y2
The y-coordinate of the second point.

See Also

DrawLine Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawLine Method (Pen, Point, Point)

Draws a line segment connecting two points.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawLine (
    Pen pen,
    Point p1,
    Point p2
)
```

Visual Basic Copy Code

```
Sub DrawLine( _
    pen As Pen, _
    p1 As Point, _
    p2 As Point _
)
```

Parameters

pen
A Pen object that determines the color, width, and style of the line.

p1
A **Microsoft.Maui.Graphics.Point** that represents the first point to connect.

p2
A **Microsoft.Maui.Graphics.Point** that represents the second point to connect.

See Also

DrawLine Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawLines Method

Draws a series of line segments that connect an array of Point structures.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawLines (
    Pen pen,
    IList<Point> points
)
```

Visual Basic Copy Code

```
Sub DrawLines( _
    pen As Pen, _
```

```
    points As IList(Of Point) _
)
```

Parameters

pen
A Pen object that determines the color, width, and style of the line segments.

points
An array of Point structures that represent the points to connect.

See Also

DrawLines Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawPath Method

Draws an IGraphicsPath object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawPath (
    Pen pen,
    IGraphicsPath path
)
```

Visual Basic Copy Code

```
Sub DrawPath( _
    pen As Pen, _
    path As IGraphicsPath _
)
```

Parameters

pen
A Pen object that determines the color, width, and style of the path.

path
An IGraphicsPath object to draw.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawPoint Method

Draws a point at the specified coordinates.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawPoint (
    SolidBrush brush,
    double x,
```

```
    double y,
    double size
)
```

Visual Basic Copy Code

```
Sub DrawPoint( _
    brush As SolidBrush, _
    x As Double, _
    y As Double, _
    size As Double _
)
```

Parameters

brush
A SolidBrush object that determines the color of the point.

x
The x-coordinate of the point.

y
The y-coordinate of the point.

size
The points' size.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawPolygon Method

Draws a polygon specified by a poins list.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawPolygon (
    Pen pen,
    IList<Point> points
)
```

Visual Basic Copy Code

```
Sub DrawPolygon( _
    pen As Pen, _
    points As IList(Of Point) _
)
```

Parameters

pen
A Pen object that determines the color, width, and style of the polygon.

points
A list of points that represents the polygon to draw.

See Also

DrawPolygon Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawRectangle Method

Draws a rectangle specified by a coordinate pair, a width, and a height.

Overload List

Name	Description
IGraphics.DrawRectangle (Pen, Double, Double, Double, Double)	Draws a rectangle specified by a coordinate pair, a width, and a height.
IGraphics.DrawRectangle (Pen, Rect)	Draws a rectangle specified by a **Rect** structure.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawRectangle Method (Pen, Double, Double, Double, Double)

Draws a rectangle specified by a coordinate pair, a width, and a height.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawRectangle (
    Pen pen,
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Sub DrawRectangle( _
    pen As Pen, _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
)
```

Parameters

pen
A Pen object that determines the color, width, and style of the rectangle.

x
The x-coordinate of the upper-left corner of the rectangle to draw.

y
The y-coordinate of the upper-left corner of the rectangle to draw.

width
The width of the rectangle to draw.

height
The height of the rectangle to draw.

See Also

DrawRectangle Method Overload List
IGraphics Members

IGraphics.DrawRectangle Method (Pen, Rect)

Draws a rectangle specified by a **Rect** structure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawRectangle (
    Pen pen,
    Rect rect
)
```

Visual Basic Copy Code

```
Sub DrawRectangle( _
    pen As Pen, _
    rect As Rect _
)
```

Parameters

pen
A Pen object that determines the color, width, and style of the rectangle.

rect
A **Microsoft.Maui.Graphics.Rect** that represents the rectangle to draw.

See Also

DrawRectangle Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawString Method

Draws the specified text string at the specified location with the specified **Brush** and **Font** objects.

Overload List

Name	Description
IGraphics.DrawString (String, Font, Brush, Double, Double)	Draws the specified text string at the specified location with the specified **Brush** and **Font** objects.
IGraphics.DrawString (String, Font, Brush, Point, StringFormat)	Draws the specified text string at the specified location with the specified **Brush** and **Font** objects using the formatting attributes of the specified **StringFormat** object.
IGraphics.DrawString (String, Font, Brush, Rect)	Draws the specified text string in the specified rectangle with the specified **Brush** and **Font** objects.
IGraphics.DrawString (String, Font, Brush, Rect, StringFormat)	Draws the specified text string in the specified rectangle with the specified **Brush** and **Font** objects using the formatting attributes of the specified **StringFormat** object.

See Also

IGraphics.DrawString Method (String, Font, Brush, Double, Double)

Draws the specified text string at the specified location with the specified **Brush** and **Font** objects.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawString (
    string text,
    Font font,
    Brush brush,
    double x,
    double y
)
```

Visual Basic Copy Code

```
Sub DrawString( _
    text As String, _
    font As Font, _
    brush As Brush, _
    x As Double, _
    y As Double _
)
```

Parameters

text
The string to draw.

font
A Font object that defines the text format of the string.

brush
A Brush object that determines the color and texture of the drawn text.

x
The x-coordinate of the upper-left corner of the drawn text.

y
The y-coordinate of the upper-left corner of the drawn text.

See Also

DrawString Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawString Method (String, Font, Brush, Point, StringFormat)

Draws the specified text string at the specified location with the specified **Brush** and **Font** objects using the formatting attributes of the specified **StringFormat** object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawString (
    string text,
    Font font,
    Brush brush,
    Point location,
    StringFormat format
)
```

Visual Basic Copy Code

```
Sub DrawString( _
    text As String, _
    font As Font, _
    brush As Brush, _
    location As Point, _
    format As StringFormat _
)
```

Parameters

text
The string to draw.

font
A Font object that defines the text format of the string.

brush
A Brush object that determines the color and texture of the drawn text.

location
A Point structure that specifies the upper-left corner of the drawn text.

format
A StringFormat object that specifies formatting attributes, such as line spacing and alignment, that are applied to the drawn text.

See Also

DrawString Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawString Method (String, Font, Brush, Rect)

Draws the specified text string in the specified rectangle with the specified **Brush** and **Font** objects.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawString (
    string text,
    Font font,
    Brush brush,
    Rect rect
)
```

Visual Basic Copy Code

```
Sub DrawString( _
    text As String, _
    font As Font, _
    brush As Brush, _
    rect As Rect _
)
```

Parameters

text
The string to draw.

font
A Font object that defines the text format of the string.

brush
A Brush object that determines the color and texture of the drawn text.

rect
A **Microsoft.Maui.Graphics.Rect** that specifies the location of the drawn text.

See Also

DrawString Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DrawString Method (String, Font, Brush, Rect, StringFormat)

Draws the specified text string in the specified rectangle with the specified **Brush** and **Font** objects using the formatting attributes of the specified **StringFormat** object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void DrawString (
    string text,
    Font font,
    Brush brush,
    Rect rect,
    StringFormat format
)
```

Visual Basic Copy Code

```
Sub DrawString( _
    text As String, _
    font As Font, _
    brush As Brush, _
    rect As Rect, _
    format As StringFormat _
)
```

Parameters

text
The string to draw.

font
A Font object that defines the text format of the string.

brush
A Brush object that determines the color and texture of the drawn text.

rect
A **Microsoft.Maui.Graphics.Rect** that specifies the location of the drawn text.

format
A StringFormat object that specifies formatting attributes, such as line spacing and alignment, that are applied to the drawn text.

See Also

DrawString Method Overload List
IGraphics Members

IGraphics.EndHyperLink Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void EndHyperLink ()
```

Visual Basic Copy Code

```
Sub EndHyperLink ()
```

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.FillEllipse Method

Fills the interior of an ellipse defined by a bounding rectangle specified by a pair of coordinates, a width, and a height.

Overload List

Name	Description
IGraphics.FillEllipse (Brush, Double, Double, Double, Double)	Fills the interior of an ellipse defined by a bounding rectangle specified by a pair of coordinates, a width, and a height.
IGraphics.FillEllipse (Brush, Rect)	Fills the interior of an ellipse defined by a bounding rectangle specified by a **Rect** structure.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.FillEllipse Method (Brush, Double, Double, Double, Double)

Fills the interior of an ellipse defined by a bounding rectangle specified by a pair of coordinates, a width, and a height.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void FillEllipse (
    Brush brush,
    double x,
    double y,
    double width,
```

```
    double height
)
```

Visual Basic Copy Code

```
Sub FillEllipse( _
    brush As Brush, _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
)
```

Parameters

brush

A Brush object that determines the characteristics of the fill.

x

The x-coordinate of the upper-left corner of the bounding rectangle that defines the ellipse.

y

The y-coordinate of the upper-left corner of the bounding rectangle that defines the ellipse.

width

The width of the bounding rectangle that defines the ellipse.

height

The height of the bounding rectangle that defines the ellipse.

See Also

FillEllipse Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.FillEllipse Method (Brush, Rect)

Fills the interior of an ellipse defined by a bounding rectangle specified by a **Rect** structure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void FillEllipse (
    Brush brush,
    Rect rect
)
```

Visual Basic Copy Code

```
Sub FillEllipse( _
    brush As Brush, _
    rect As Rect _
)
```

Parameters

brush

A Brush object that specifies how to fill the ellipse.

rect

A **Microsoft.Maui.Graphics.Rect** that represents the bounding rectangle that defines the ellipse.

See Also

FillEllipse Method Overload List
IGraphics Members

IGraphics.FillPath Method

Fills the interior of a **GraphicsPath** object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void FillPath (
    Brush brush,
    IGraphicsPath path
)
```

Visual Basic Copy Code

```
Sub FillPath( _
    brush As Brush, _
    path As IGraphicsPath _
)
```

Parameters

brush
A Brush object that determines the characteristics of the fill.

path
An IGraphicsPath object that represents the path to fill.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.FillPolygon Method

Fills the interior of a polygon specified by a points list.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void FillPolygon (
    Brush brush,
    IList<Point> points
)
```

Visual Basic Copy Code

```
Sub FillPolygon( _
    brush As Brush, _
    points As IList(Of Point) _
)
```

Parameters

brush
A Brush object that determines the characteristics of the fill.

points

A list of points that represents the polygon to fill.

See Also

FillPolygon Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.FillRectangle Method

Fills the interior of a rectangle specified by a pair of coordinates, a width, and a height.

Overload List

Name	Description
IGraphics.FillRectangle (Brush, Double, Double, Double, Double)	Fills the interior of a rectangle specified by a pair of coordinates, a width, and a height.
IGraphics.FillRectangle (Brush, Rect)	Fills the interior of a rectangle specified by a **Rect** structure.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.FillRectangle Method (Brush, Double, Double, Double, Double)

Fills the interior of a rectangle specified by a pair of coordinates, a width, and a height.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void FillRectangle (
    Brush brush,
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Sub FillRectangle( _
    brush As Brush, _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
)
```

Parameters

brush

A Brush object that determines the characteristics of the fill.

x

The x-coordinate of the upper-left corner of the rectangle to fill.

y

The y-coordinate of the upper-left corner of the rectangle to fill.

width
The width of the rectangle to fill.

height
The height of the rectangle to fill.

See Also

FillRectangle Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.FillRectangle Method (Brush, Rect)

Fills the interior of a rectangle specified by a **Rect** structure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void FillRectangle (
    Brush brush,
    Rect rect
)
```

Visual Basic Copy Code

```
Sub FillRectangle( _
    brush As Brush, _
    rect As Rect _
)
```

Parameters

brush
A Brush object that specifies how to fill the rectangle.

rect
A **Microsoft.Maui.Graphics.Rect** that represents the rectangle to fill.

See Also

FillRectangle Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.GetFontHeight Method

Gets the height of the specified font.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
double GetFontHeight (
    Font font
)
```

Visual Basic Copy Code

```
Function GetFontHeight( _
    font As Font _
) As Double
```

Parameters

font
The font to measure.

Return Value

The height of the font.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.GetImageSize Method

Returns the physical size of the specified image.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
Size GetImageSize (
    IImage image
)
```

Visual Basic Copy Code

```
Function GetImageSize( _
    image As IImage _
) As Size
```

Parameters

image
The image to test.

Return Value

The size of the image.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.MeasureString Method

Measures the specified string when drawn with the specified **Font** object.

Overload List

Name	Description
IGraphics.MeasureString (String, Font)	Measures the specified string when drawn with the specified Font object.
IGraphics.MeasureString (String, Font, Double, StringFormat)	Measures the specified string when drawn with the specified **Font** object and formatted with the specified StringFormat object.

IGraphics.MeasureString (String, Font, Size, StringFormat, Int32@, Int32@)	Measures the specified string when drawn with the specified **Font** object and formatted with the specified **StringFormat** object.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.MeasureString Method (String, Font)

Measures the specified string when drawn with the specified Font object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
Size MeasureString (
    string text,
    Font font
)
```

Visual Basic Copy Code

```
Function MeasureString( _
    text As String, _
    font As Font _
) As Size
```

Parameters

text
The string to measure.

font
A Font object that defines the text format of the string.

Return Value

This method returns a **Microsoft.Maui.Graphics.Size** that represents the size, in pixels, of the string specified by the *text* parameter as drawn with the *font* parameter.

See Also

MeasureString Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.MeasureString Method (String, Font, Double, StringFormat)

Measures the specified string when drawn with the specified Font object and formatted with the specified StringFormat object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
Size MeasureString (
    string text,
    Font font,
    double maxWidth,
```

```
    StringFormat format
)
```

Visual Basic Copy Code

```
Function MeasureString( _
    text As String, _
    font As Font, _
    maxWidth As Double, _
    format As StringFormat _
) As Size
```

Parameters

text

The string to measure.

font

A Font object that defines the text format of the string.

maxWidth

The maximum width of the string.

format

A StringFormat object that represents formatting information, such as line spacing, for the string.

Return Value

This method returns a **Microsoft.Maui.Graphics.Size** that represents the size, in pixels, of the string specified in the *text* parameter as drawn with the *font* parameter and the *stringFormat* parameter.

See Also

MeasureString Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.MeasureString Method (String, Font, Size, StringFormat, Int32@, Int32@)

Measures the specified string when drawn with the specified Font object and formatted with the specified StringFormat object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
Size MeasureString (
    string text,
    Font font,
    Size layoutArea,
    StringFormat stringFormat,
    out int charactersFitted,
    out int linesFilled
)
```

Visual Basic Copy Code

```
Function MeasureString( _
    text As String, _
    font As Font, _
    layoutArea As Size, _
    stringFormat As StringFormat, _
    <System.Runtime.InteropServices.Out()> ByRef charactersFitted As Integer, _
    <System.Runtime.InteropServices.Out()> ByRef linesFilled As Integer _
) As Size
```

Parameters

text

The string to measure.

font

A Font object that defines the text format of the string.

layoutArea

A **Microsoft.Maui.Graphics.Size** that specifies the maximum layout area for the text.

stringFormat

A StringFormat object that represents formatting information, such as line spacing, for the string.

charactersFitted

Number of characters in the string.

linesFilled

Number of text lines in the string.

Return Value

This method returns a **Microsoft.Maui.Graphics.Size** that represents the size of the string, in pixels, of the *text* parameter as drawn with the *font* parameter and the *stringFormat* parameter.

See Also

MeasureString Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.MultiplyTransform Method

Multiplies the world transformation of this IGraphics object and specified the **Matrix** object in the specified order.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void MultiplyTransform (
    IMatrix m,
    MatrixOrder order = MatrixOrder.Append
)
```

Visual Basic Copy Code

```
Sub MultiplyTransform( _
    m As IMatrix, _
    Optional order As MatrixOrder = MatrixOrder.Append _
)
```

Parameters

m

The multipler matrix.

order

Optional. Member of the MatrixOrder enumeration that determines the order of the multiplication.

See Also

MultiplyTransform Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.PopOpacity Method

Restores global opacity.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void PopOpacity ()
```

Visual Basic Copy Code

```
Sub PopOpacity ()
```

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.PushOpacity Method

Sets global opacity.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void PushOpacity (
    double opacity
)
```

Visual Basic Copy Code

```
Sub PushOpacity( _
    opacity As Double _
)
```

Parameters

opacity

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.ResetTransform Method

Resets the world transformation matrix of this IGraphics object to the identity matrix.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void ResetTransform ()
```

Visual Basic Copy Code

```
Sub ResetTransform ()
```

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.Restore Method

Restores the state of this IGraphics object to the specified state argument.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void Restore (
    Object state
)
```

Visual Basic Copy Code

```
Sub Restore( _
    state As Object _
)
```

Parameters

state
An object that represents the state to which to restore this IGraphics object.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.RotateTransform Method

Applies the specified rotation to the transformation matrix of this IGraphics object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void RotateTransform (
    double angle
)
```

Visual Basic Copy Code

```
Sub RotateTransform( _
    angle As Double _
)
```

Parameters

angle
The angle of rotation in degrees.

See Also

RotateTransform Method Overload List
IGraphics Members

IGraphics.Save Method

Saves the current state of this IGraphics object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
Object Save ()
```

Visual Basic Copy Code

```
Function Save () As Object
```

Return Value

An object that represents the saved state of this IGraphics object.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.ScaleTransform Method

Applies the specified scaling operation to the transformation matrix of this IGraphics object by prepending it to the object's transformation matrix.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void ScaleTransform (
    double sx,
    double sy
)
```

Visual Basic Copy Code

```
Sub ScaleTransform( _
    sx As Double, _
    sy As Double _
)
```

Parameters

sx
The scale factor in the x direction.

sy
The scale factor in the y direction.

See Also

ScaleTransform Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.SetClip Method

Sets the clipping region of this IGraphics object to the Clip property of the specified **IGraphics** object.

Overload List

Name	Description
IGraphics.SetClip (IGraphicsPath)	Sets the clipping region of this **IGraphics** object to the specified IGraphicsPath object.
IGraphics.SetClip (IGraphicsPath, CombineMode)	Sets the clipping region of this **IGraphics** object to the result of the specified operation combining the current clip region and the specified **GraphicsPath** object.
IGraphics.SetClip (Rect)	Sets the clipping region of this **IGraphics** object to the rectangle specified by a **Rect** structure.
IGraphics.SetClip (Rect, CombineMode)	Sets the clipping region of this **IGraphics** object to the result of the specified operation combining the current clip region and the rectangle specified by a **Rect** structure.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.SetClip Method (IGraphicsPath)

Sets the clipping region of this IGraphics object to the specified IGraphicsPath object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void SetClip (
    IGraphicsPath path
)
```

Visual Basic Copy Code

```
Sub SetClip( _
    path As IGraphicsPath _
)
```

Parameters

path
An IGraphicsPath object that represents the new clip region.

See Also

SetClip Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.SetClip Method (IGraphicsPath, CombineMode)

Sets the clipping region of this IGraphics object to the result of the specified operation combining the current clip region and the specified **GraphicsPath** object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void SetClip (
    IGraphicsPath path,
    CombineMode mode
)
```

Visual Basic Copy Code

```
Sub SetClip( _
    path As IGraphicsPath, _
    mode As CombineMode _
)
```

Parameters

path

A **GraphicsPath** object to combine.

mode

Member of the CombineMode enumeration that specifies the combining operation to use.

See Also

SetClip Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.SetClip Method (Rect)

Sets the clipping region of this IGraphics object to the rectangle specified by a **Rect** structure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void SetClip (
    Rect rect
)
```

Visual Basic Copy Code

```
Sub SetClip( _
    rect As Rect _
)
```

Parameters

rect

A **Microsoft.Maui.Graphics.Rect** that represents the new clip region.

See Also

SetClip Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.SetClip Method (Rect, CombineMode)

Sets the clipping region of this IGraphics object to the result of the specified operation combining the current clip region and the rectangle specified by a **Rect** structure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void SetClip (
    Rect rect,
    CombineMode mode
)
```

Visual Basic Copy Code

```
Sub SetClip( _
    rect As Rect, _
    mode As CombineMode _
)
```

Parameters

rect

A **Microsoft.Maui.Graphics.Rect** to combine.

mode

Member of the CombineMode enumeration that specifies the combining operation to use.

See Also

SetClip Method Overload List
IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.SetTransform Method

Sets the current transformation.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void SetTransform (
    IMatrix m
)
```

Visual Basic Copy Code

```
Sub SetTransform( _
    m As IMatrix _
)
```

Parameters

m

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.TranslateTransform Method

Prepends the specified translation to the transformation matrix of this IGraphics object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void TranslateTransform (
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Sub TranslateTransform( _
    dx As Double, _
    dy As Double _
)
```

Parameters

dx
The x component of the translation.

dy
The y component of the translation.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics Properties

IGraphics.Clip Property

Gets or sets an IGraphicsPath object that limits the drawing region of this IGraphics object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
IGraphicsPath Clip { get; }
```

Visual Basic Copy Code

```
ReadOnly Property Clip As IGraphicsPath
```

Property Value

This property specifies an IGraphicsPath object that limits the portion of this IGraphics object that is currently available for drawing.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DpiX Property

Gets the horizontal resolution of this IGraphics object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
double DpiX { get; set; }
```

Visual Basic Copy Code

```
Property DpiX As Double
```

Property Value

This property indicates a value, in dots per inch, for the horizontal resolution supported by this IGraphics object.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.DpiY Property

Gets the vertical resolution of this IGraphics object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
double DpiY { get; set; }
```

Visual Basic Copy Code

```
Property DpiY As Double
```

Property Value

This property indicates a value, in dots per inch, for the vertical resolution supported by this IGraphics object.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.EnableAntialiasing Property

Gets or sets whether to anti-alias the graphics primitives.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
bool EnableAntialiasing { get; set; }
```

Visual Basic Copy Code

```
Property EnableAntialiasing As Boolean
```

Property Value

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.SupportsPenWidth0 Property

Gets a value indicating whether the IGraphics object supports pens with zero width.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
bool SupportsPenWidth0 { get; }
```

Visual Basic Copy Code

```
ReadOnly Property SupportsPenWidth0 As Boolean
```

Property Value

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.Transform Property

Gets or sets the world transformation for this IGraphics object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
IMatrix Transform { get; }
```

Visual Basic Copy Code

```
ReadOnly Property Transform As IMatrix
```

Property Value

This property specifies a Matrix object that represents the world transformation for this IGraphics object.

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphics.UnitScale Property

Gets the scale for current unit.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
double UnitScale { get; set; }
```

Visual Basic Copy Code

```
Property UnitScale As Double
```

Property Value

See Also

IGraphics Members
IGraphics Interface
MindFusion.Drawing Namespace

IGraphicsPath Interface

Represents a series of connected lines and curves.

Syntax

C# Copy Code

```
public interface IGraphicsPath
```

Visual Basic Copy Code

```
Public Interface IGraphicsPath
```

See Also

IGraphicsPath Members
MindFusion.Drawing Namespace

IGraphicsPath Members

Represents a series of connected lines and curves.

The following tables list the members exposed by the IGraphicsPath type.

Public Properties

	Name	Description
	FillMode	Gets or sets the path fill mode.
	PathPoints	Gets the points in the path.
	PointCount	Gets the number of points in the path.

Public Methods

	Name	Description

	AddArc	Overloaded.
	AddBezier	Overloaded.
	AddBeziers	Adds a sequence of connected cubic Bézier curves to the current figure.
	AddEllipse	Overloaded.
	AddLine	Overloaded.
	AddLines	Appends a series of connected line segments to the end of this IGraphicsPath.
	AddPath	Appends specified path to this path.
	AddPolygon	Adds a polygon to the current figure.
	AddRectangle	Overloaded.
	AddRoundRect	Adds a rounded rectangle to this path.
	AddString	Adds a string to this path.
	Clone	Clones this GraphicsPath.
	CloseFigure	Closes the current figure and starts a new figure. If the current figure contains a sequence of connected lines and curves, the method closes the loop by connecting a line from the endpoint to the starting point.
	Dispose	Disposes the path.
	Flatten	Overloaded.
	GetBounds	Returns a rectangle that bounds this IGraphicsPath.
	GetLastPoint	Gets the current last point in the path, if any, otherwise, null.
	IntersectsWith	Checks whether this IGraphicsPath intersects with the specified IGraphicsPath.
	IsVisible	Indicates whether the specified point is contained within this IGraphicsPath.
	StartFigure	Starts a new figure without closing the current figure. All subsequent points added to the path are added to this new figure.
	Transform	Applies a transform matrix to this IGraphicsPath.
	Widen	Widens the path.

See Also

IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath Methods

IGraphicsPath.AddArc Method

Overload List

Name	Description
IGraphicsPath.AddArc (Double, Double, Double, Double, Double, Double)	Appends an elliptical arc to the current figure.

IGraphicsPath.AddArc (Rect, Double, Double)	Appends an elliptical arc to the current figure.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddArc Method (Double, Double, Double, Double, Double, Double)

Appends an elliptical arc to the current figure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddArc (
    double x,
    double y,
    double width,
    double height,
    double startAngle,
    double sweepAngle
)
```

Visual Basic Copy Code

```
Sub AddArc( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double, _
    startAngle As Double, _
    sweepAngle As Double _
)
```

Parameters

x
The x-coordinate of the upper-left corner of the rectangular region that defines the ellipse from which the arc is drawn.

y
The y-coordinate of the upper-left corner of the rectangular region that defines the ellipse from which the arc is drawn.

width
The width of the rectangular region that defines the ellipse from which the arc is drawn.

height
The height of the rectangular region that defines the ellipse from which the arc is drawn.

startAngle
The starting angle of the arc, measured in degrees clockwise from the x-axis.

sweepAngle
The angle between startAngle and the end of the arc.

See Also

AddArc Method Overload List
IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddArc Method (Rect, Double, Double)

Appends an elliptical arc to the current figure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddArc (
    Rect rect,
    double startAngle,
    double sweepAngle
)
```

Visual Basic Copy Code

```
Sub AddArc( _
    rect As Rect, _
    startAngle As Double, _
    sweepAngle As Double _
)
```

Parameters

rect
A Rectangle that represents the rectangular bounds of the ellipse from which the arc is taken.

startAngle
The starting angle of the arc, measured in degrees clockwise from the x-axis.

sweepAngle
The angle between startAngle and the end of the arc.

See Also

AddArc Method Overload List
IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddBezier Method

Overload List

Name	Description
IGraphicsPath.AddBezier (Double, Double, Double, Double, Double, Double, Double, Double)	Adds a cubic Bézier curve to the current figure.
IGraphicsPath.AddBezier (Point, Point, Point, Point)	Adds a cubic Bézier curve to the current figure.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddBezier Method (Double, Double, Double, Double, Double, Double, Double, Double)

Adds a cubic Bézier curve to the current figure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddBezier (
    double x1,
    double y1,
    double x2,
    double y2,
    double x3,
    double y3,
    double x4,
    double y4
)
```

Visual Basic Copy Code

```
Sub AddBezier( _
    x1 As Double, _
    y1 As Double, _
    x2 As Double, _
    y2 As Double, _
    x3 As Double, _
    y3 As Double, _
    x4 As Double, _
    y4 As Double _
)
```

Parameters

x1
The x-coordinate of the starting point of the curve.

y1
The y-coordinate of the starting point of the curve.

x2
The x-coordinate of the first control point for the curve.

y2
The y-coordinate of the first control point for the curve.

x3
The x-coordinate of the second control point for the curve.

y3
The y-coordinate of the second control point for the curve.

x4
The x-coordinate of the endpoint of the curve.

y4
The y-coordinate of the endpoint of the curve.

See Also

AddBezier Method Overload List
IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddBezier Method (Point, Point, Point, Point)

Adds a cubic Bézier curve to the current figure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddBezier (
    Point p1,
    Point p2,
    Point p3,
    Point p4
)
```

Visual Basic Copy Code

```
Sub AddBezier( _
    p1 As Point, _
    p2 As Point, _
    p3 As Point, _
    p4 As Point _
)
```

Parameters

p1
A Point that represents the starting point of the curve.

p2
A Point that represents the first control point for the curve.

p3
A Point that represents the second control point for the curve.

p4
A Point that represents the endpoint of the curve.

See Also

AddBezier Method Overload List
IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddBeziers Method

Adds a sequence of connected cubic Bézier curves to the current figure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddBeziers (
    IList<Point> points
)
```

Visual Basic Copy Code

```
Sub AddBeziers( _
    points As IList(Of Point) _
)
```

Parameters

points
An array of Point structures that represents the points that define the curves.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddEllipse Method

Overload List

Name	Description
IGraphicsPath.AddEllipse (Double, Double, Double, Double)	Adds an ellipse to the current path.
IGraphicsPath.AddEllipse (Rect)	Adds an ellipse to the current path.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddEllipse Method (Double, Double, Double, Double)

Adds an ellipse to the current path.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddEllipse (
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Sub AddEllipse( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
)
```

Parameters

x
The x-coordinate of the upper-left corner of the bounding rectangle that defines the ellipse.

y
The y-coordinate of the upper-left corner of the bounding rectangle that defines the ellipse.

width
The width of the bounding rectangle that defines the ellipse.

height
The height of the bounding rectangle that defines the ellipse.

See Also

AddEllipse Method Overload List
IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddEllipse Method (Rect)

Adds an ellipse to the current path.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddEllipse (
    Rect rect
)
```

Visual Basic Copy Code

```
Sub AddEllipse( _
    rect As Rect _
)
```

Parameters

rect
A Rectangle that represents the bounding rectangle that defines the ellipse.

See Also

AddEllipse Method Overload List
IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddLine Method

Overload List

Name	Description
IGraphicsPath.AddLine (Double, Double, Double, Double)	Appends a line segment to the current figure.
IGraphicsPath.AddLine (Point, Point)	Appends a line segment to this IGraphicsPath.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddLine Method (Double, Double, Double, Double)

Appends a line segment to the current figure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

Copy Code

```
void AddLine (
    double x1,
    double y1,
    double x2,
    double y2
)
```

Visual Basic

Copy Code

```
Sub AddLine( _
    x1 As Double, _
    y1 As Double, _
    x2 As Double, _
    y2 As Double _
)
```

Parameters

x1
The x-coordinate of the starting point of the line.

y1
The y-coordinate of the starting point of the line.

x2
The x-coordinate of the endpoint of the line.

y2
The y-coordinate of the endpoint of the line.

See Also

AddLine Method Overload List
IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddLine Method (Point, Point)

Appends a line segment to this IGraphicsPath.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

Copy Code

```
void AddLine (
    Point p1,
    Point p2
)
```

Visual Basic

Copy Code

```
Sub AddLine( _
    p1 As Point, _
    p2 As Point _
)
```

Parameters

p1
A Point that represents the starting point of the line.

p2
A Point that represents the endpoint of the line.

See Also

AddLine Method Overload List
IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddLines Method

Appends a series of connected line segments to the end of this IGraphicsPath.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddLines (
    IList<Point> points
)
```

Visual Basic Copy Code

```
Sub AddLines( _
    points As IList(Of Point) _
)
```

Parameters

points
A list of Point structures that represents the points that define the line segments to add.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddPath Method

Appends specified path to this path.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddPath (
    IGraphicsPath path,
    bool connect = true
)
```

Visual Basic Copy Code

```
Sub AddPath( _
    path As IGraphicsPath, _
    Optional connect As Boolean = True _
)
```

Parameters

path

connect
Optional.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddPolygon Method

Adds a polygon to the current figure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddPolygon (
    IList<Point> points
)
```

Visual Basic Copy Code

```
Sub AddPolygon( _
    points As IList(Of Point) _
)
```

Parameters

points
A list of points that define the polygon.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddRectangle Method

Overload List

Name	Description
IGraphicsPath.AddRectangle (Double, Double, Double, Double)	Adds a rectangle to this path.
IGraphicsPath.AddRectangle (Rect)	Adds a rectangle to this path.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddRectangle Method (Double, Double, Double, Double)

Adds a rectangle to this path.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddRectangle (
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Sub AddRectangle( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
)
```

Parameters

x
The x-coordinate of the upper-left corner of the rectangle to add.

y
The y-coordinate of the upper-left corner of the rectangle to add.

width
The width of the rectangle to add.

height
The height of the rectangle to add.

See Also

AddRectangle Method Overload List
IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddRectangle Method (Rect)

Adds a rectangle to this path.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddRectangle (
    Rect rect
)
```

Visual Basic Copy Code

```
Sub AddRectangle( _
    rect As Rect _
)
```

Parameters

rect
The rectangle to add.

See Also

AddRectangle Method Overload List
IGraphicsPath Members

IGraphicsPath.AddRoundRect Method

Adds a rounded rectangle to this path.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddRoundRect (
    double x,
    double y,
    double width,
    double height,
    double radius
)
```

Visual Basic Copy Code

```
Sub AddRoundRect( _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double, _
    radius As Double _
)
```

Parameters

x
The x-coordinate of the upper-left corner of the rectangle to add.

y
The y-coordinate of the upper-left corner of the rectangle to add.

width
The width of the rectangle to add.

height
The height of the rectangle to add.

radius
The roundness of the rectangle corners.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.AddString Method

Adds a string to this path.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void AddString (
    string text,
    Font font,
    Point point,
```

```
    StringFormat format
)
```

Visual Basic Copy Code

```
Sub AddString( _
    text As String, _
    font As Font, _
    point As Point, _
    format As StringFormat _
)
```

Parameters

text

font

point

format

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.Clone Method

Clones this GraphicsPath.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
IGraphicsPath Clone ()
```

Visual Basic Copy Code

```
Function Clone () As IGraphicsPath
```

Return Value

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.CloseFigure Method

Closes the current figure and starts a new figure. If the current figure contains a sequence of connected lines and curves, the method closes the loop by connecting a line from the endpoint to the starting point.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void CloseFigure ()
```

Visual Basic Copy Code

```
Sub CloseFigure ()
```

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.Dispose Method

Disposes the path.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void Dispose ()
```

Visual Basic Copy Code

```
Sub Dispose ()
```

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.Flatten Method

Overload List

Name	Description
IGraphicsPath.Flatten ()	Converts each curve in this path into a sequence of connected line segments.
IGraphicsPath.Flatten (Double)	Converts each curve in this path into a sequence of connected line segments using the specified factor.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.Flatten Method ()

Converts each curve in this path into a sequence of connected line segments.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void Flatten ()
```

Visual Basic Copy Code

```
Sub Flatten ()
```

See Also

Flatten Method Overload List
IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.Flatten Method (Double)

Converts each curve in this path into a sequence of connected line segments using the specified factor.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void Flatten (
    double flatness
)
```

Visual Basic Copy Code

```
Sub Flatten( _
    flatness As Double _
)
```

Parameters

flatness
Specifies the factor of the flatness. This value ranges between 1 and 100. Reducing the flatness value will increase the number of line segments in the approximation.

See Also

Flatten Method Overload List
IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.GetBounds Method

Returns a rectangle that bounds this IGraphicsPath.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
Rect GetBounds ()
```

Visual Basic Copy Code

```
Function GetBounds () As Rect
```

Return Value

A Rectangle that represents a rectangle that bounds this IGraphicsPath.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.GetLastPoint Method

Gets the current last point in the path, if any, otherwise, null.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
Nullable<Point> GetLastPoint ()
```

Visual Basic Copy Code

```
Function GetLastPoint () As Nullable(Of Point)
```

Return Value

A Point object representing the last point in the path or null.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.IntersectsWith Method

Checks whether this IGraphicsPath intersects with the specified IGraphicsPath.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
bool IntersectsWith (
    IGraphicsPath path
)
```

Visual Basic Copy Code

```
Function IntersectsWith( _
    path As IGraphicsPath _
) As Boolean
```

Parameters

path
The IGraphicsPath to test.

Return Value

true if the two paths intersect; otherwise, false.

See Also

IGraphicsPath Members
IGraphicsPath Interface

IGraphicsPath.IsVisible Method

Indicates whether the specified point is contained within this IGraphicsPath.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
bool IsVisible (
    Point point
)
```

Visual Basic Copy Code

```
Function IsVisible( _
    point As Point _
) As Boolean
```

Parameters

point
The point to test.

Return Value

true if the specified point is contained within this IGraphicsPath; otherwise, false.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.StartFigure Method

Starts a new figure without closing the current figure. All subsequent points added to the path are added to this new figure.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void StartFigure ()
```

Visual Basic Copy Code

```
Sub StartFigure ()
```

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.Transform Method

Applies a transform matrix to this IGraphicsPath.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void Transform (
    IMatrix matrix
)
```

Visual Basic Copy Code

```
Sub Transform( _
    matrix As IMatrix _
)
```

Parameters

matrix
An IMatrix that represents the transformation to apply.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.Widen Method

Widens the path.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void Widen (
    float width,
    float flatness = 0
)
```

Visual Basic Copy Code

```
Sub Widen( _
    width As Single, _
    Optional flatness As Single = 0 _
)
```

Parameters

width

flatness
Optional.

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath Properties

IGraphicsPath.FillMode Property

Gets or sets the path fill mode.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
FillMode FillMode { get; set; }
```

Visual Basic Copy Code

```
Property FillMode As FillMode
```

Property Value

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.PathPoints Property

Gets the points in the path.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
List<Point> PathPoints { get; }
```

Visual Basic Copy Code

```
ReadOnly Property PathPoints As List(Of Point)
```

Property Value

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

IGraphicsPath.PointCount Property

Gets the number of points in the path.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
int PointCount { get; }
```

Visual Basic Copy Code

```
ReadOnly Property PointCount As Integer
```

Property Value

See Also

IGraphicsPath Members
IGraphicsPath Interface
MindFusion.Drawing Namespace

ImageAlign Enumeration

Specifies the position and alignment of a picture relative to its destination rectangle.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum ImageAlign
```

Visual Basic Copy Code

```
Public Enum ImageAlign
```

Members

	Member name	Description
	Unspecified	Invalid image alignment.
	Center	The image is centered.
	Fit	The image is resized to fit the size the destination rectangle.
	Stretch	The image is stretched to fill the destination rectangle.
	Tile	The image is tiled across the destination rectangle.
	TopLeft	The image is aligned to the top left corner.
	BottomLeft	The image is aligned to the bottom left corner.
	TopRight	The image is aligned o the top right corner.
	BottomRight	The image is aligned to the bottom right corner.
	TopCenter	The image is centered horizontally and aligned to the top side.
	BottomCenter	The image is centered horizontally and aligned to the bottom side.
	MiddleLeft	The image is centered vertically and aligned to the left-hand side.
	MiddleRight	The image is centered vertically and aligned to the right-hand side.
	FitLeft	The image is resized to fit the size of the object and aligned to left side.
	FitTop	The image is resized to fit the size of the object and aligned to top side.
	FitRight	The image is resized to fit the size of the object and aligned to right side.
	FitBottom	The image is resized to fit the size of the object and aligned to bottom side.

See Also

IMatrix Interface

Encapsulates an affine matrix that represents a geometric transform.

Syntax

C# Copy Code

```
public interface IMatrix : IDisposable
```

Visual Basic Copy Code

```
Public Interface IMatrix
    Implements IDisposable
```

Inheritors

MindFusion.Drawing.MauiMatrix

See Also

IMatrix Members
MindFusion.Drawing Namespace

IMatrix Members

Encapsulates an affine matrix that represents a geometric transform.

The following tables list the members exposed by the IMatrix type.

Public Properties

	Name	Description
	Elements	
	IsIdentity	Gets a value indicating whether this IMatrix is the identity matrix.
	OffsetX	Gets the translation of this matrix along the x-axis.
	OffsetY	Gets the translation of this matrix along the y-axis.

Public Methods

	Name	Description
	Clone	Creates an exact clone of this matrix.
	Inverse	Inverts this IMatrix, if it is invertible.
	Multiply	
	Reset	Reset this matrix.
	RotateAt	Overloaded.
	Scale	Applies the specified scale vector to this IMatrix by prepending the scale vector.
	Shear	
	TransformPoint	Applies the geometric transform represented by this IMatrix to the specified point.

	TransformPoints	Applies the geometric transform represented by this IMatrix to the specified array of points.
	TransformRect	Applies the geometric transform represented by this IMatrix to the specified rectangle.
	Translate	Applies the specified translation to this IMatrix by prepending the translation.

See Also

IMatrix Interface
MindFusion.Drawing Namespace

IMatrix Methods

IMatrix.Clone Method

Creates an exact clone of this matrix.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
IMatrix Clone ()
```

Visual Basic Copy Code

```
Function Clone () As IMatrix
```

Return Value

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.Inverse Method

Inverts this IMatrix, if it is invertible.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
IMatrix Inverse ()
```

Visual Basic Copy Code

```
Function Inverse () As IMatrix
```

Return Value

An instance of the IMatrix interface.

See Also

IMatrix.Multiply Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void Multiply (
    IMatrix m,
    MatrixOrder order = MatrixOrder.Append
)
```

Visual Basic Copy Code

```
Sub Multiply( _
    m As IMatrix, _
    Optional order As MatrixOrder = MatrixOrder.Append _
)
```

Parameters

m

order
Optional.

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.Reset Method

Reset this matrix.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void Reset ()
```

Visual Basic Copy Code

```
Sub Reset ()
```

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.RotateAt Method

Overload List

Name	Description
IMatrix.RotateAt (Double)	Prepend to this IMatrix a clockwise rotation by the specified angle.
IMatrix.RotateAt (Double, Point)	Applies a clockwise rotation to this IMatrix around the point specified in the point parameter, and by prepending the rotation.

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.RotateAt Method (Double)

Prepend to this IMatrix a clockwise rotation by the specified angle.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void RotateAt (
    double angle
)
```

Visual Basic Copy Code

```
Sub RotateAt( _
    angle As Double _
)
```

Parameters

angle
The angle of the rotation, in degrees.

See Also

RotateAt Method Overload List
IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.RotateAt Method (Double, Point)

Applies a clockwise rotation to this IMatrix around the point specified in the point parameter, and by prepending the rotation.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void RotateAt (
    double angle,
    Point point
)
```

Visual Basic Copy Code

```
Sub RotateAt( _
    angle As Double, _
    point As Point _
)
```

Parameters

angle

The angle of the rotation, in degrees.

point

A Point that represents the center of the rotation.

See Also

RotateAt Method Overload List
IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.Scale Method

Applies the specified scale vector to this IMatrix by prepending the scale vector.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void Scale (
    double sx,
    double sy
)
```

Visual Basic Copy Code

```
Sub Scale( _
    sx As Double, _
    sy As Double _
)
```

Parameters

sx

The value by which to scale this IMatrix in the x-axis direction.

sy

The value by which to scale this IMatrix in the y-axis direction.

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.Shear Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
void Shear (
    double sx,
    double sy
)
```

Visual Basic Copy Code

```
Sub Shear( _
    sx As Double, _
    sy As Double _
)
```

Parameters

sx

sy

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.TransformPoint Method

Applies the geometric transform represented by this IMatrix to the specified point.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
Point TransformPoint (
    Point point
)
```

Visual Basic Copy Code

```
Function TransformPoint( _
    point As Point _
) As Point
```

Parameters

point
The point to transform.

Return Value

The transformed point.

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.TransformPoints Method

Applies the geometric transform represented by this IMatrix to the specified array of points.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
void TransformPoints (
    IList<Point> points
)
```

Visual Basic

```
Sub TransformPoints( _
    points As IList(Of Point) _
)
```

Parameters

points
A list of Point structures that represents the points to transform.

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.TransformRect Method

Applies the geometric transform represented by this IMatrix to the specified rectangle.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
Rect TransformRect (
    Rect rect
)
```

Visual Basic

```
Function TransformRect( _
    rect As Rect _
) As Rect
```

Parameters

rect
The rectangle to transform.

Return Value

The transformed rectangle.

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.Translate Method

Applies the specified translation to this IMatrix by prepending the translation.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
void Translate (
    double dx,
    double dy
)
```

Visual Basic

```
Sub Translate( _
    dx As Double, _
    dy As Double _
)
```

Parameters

dx
The x value by which to translate this IMatrix.

dy
The y value by which to translate this IMatrix.

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix Properties

IMatrix.Elements Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
float[] Elements { get; }
```

Visual Basic

```
ReadOnly Property Elements As Single()
```

Property Value

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.IsIdentity Property

Gets a value indicating whether this IMatrix is the identity matrix.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
bool IsIdentity { get; }
```

Visual Basic Copy Code

```
ReadOnly Property IsIdentity As Boolean
```

Property Value

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.OffsetX Property

Gets the translation of this matrix along the x-axis.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
double OffsetX { get; }
```

Visual Basic Copy Code

```
ReadOnly Property OffsetX As Double
```

Property Value

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

IMatrix.OffsetY Property

Gets the translation of this matrix along the y-axis.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
double OffsetY { get; }
```

Visual Basic Copy Code

```
ReadOnly Property OffsetY As Double
```

Property Value

See Also

IMatrix Members
IMatrix Interface
MindFusion.Drawing Namespace

LinearGradientBrush Class

LinearGradientBrush objects represent brushes that fill the interiors of geometric shapes with a linear gradient.

Syntax

C# Copy Code

```
public class LinearGradientBrush : Brush
```

Visual Basic Copy Code

```
Public Class LinearGradientBrush
    Inherits Brush
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.Brush
    MindFusion.Drawing.LinearGradientBrush

See Also

LinearGradientBrush Members
MindFusion.Drawing Namespace

LinearGradientBrush Members

LinearGradientBrush objects represent brushes that fill the interiors of geometric shapes with a linear gradient.

The following tables list the members exposed by the **LinearGradientBrush** type.

Public Constructors

	Name	Description
	LinearGradientBrush	Overloaded. Initializes a new instance of the **LinearGradientBrush** class.

Public Properties

	Name	Description
	Angle	Gets or sets the angle of the gradient's orientation line.
	Colors	Gets or sets an array of colors that represents the colors to use at corresponding positions along a gradient.
	EndPoint	
	Positions	Gets or sets the positions along a gradient line.
	StartPoint	
	Transform	
	WrapMode	

Public Methods

	Name	Description
	Clone	Creates an exact copy of this brush.
	Equals	Brush.Equals override.
	GetHashCode	Brush.GetHashCode override.

	ToColor	Brush.ToColor override.

See Also

LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush Constructor

Initializes a new instance of the LinearGradientBrush class.

Overload List

Name	Description
LinearGradientBrush (Color, Color, Double)	Initializes a new instance of the **LinearGradientBrush** class with the specified start and end colors and an orientation angle measured in degrees.
LinearGradientBrush (Double[], Color[], Double)	Initializes a new instance of the LinearGradientBrush class.
LinearGradientBrush (Point, Point, Color, Color)	Initializes a new instance of the LinearGradientBrush class.

See Also

LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush Constructor (Color, Color, Double)

Initializes a new instance of the LinearGradientBrush class with the specified start and end colors and an orientation angle measured in degrees.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public LinearGradientBrush (
    Color startColor,
    Color endColor,
    double angle
)
```

Visual Basic Copy Code

```
Public New ( _
    startColor As Color, _
    endColor As Color, _
    angle As Double _
)
```

Parameters

startColor
The start color of the gradient.

endColor
The end color of the gradient.

angle
The angle of the gradient.

See Also

LinearGradientBrush Constructor Overload List
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush Constructor (Double[], Color[], Double)

Initializes a new instance of the LinearGradientBrush class.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public LinearGradientBrush (
    double[] positions,
    Color[] colors,
    double angle
)
```

Visual Basic Copy Code

```
Public New ( _
    positions() As Double, _
    colors() As Color, _
    angle As Double _
)
```

Parameters

positions
The positions of the colors of the gradient.

colors
The colors of the gradient.

angle
The angle of the gradient.

See Also

LinearGradientBrush Constructor Overload List
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush Constructor (Point, Point, Color, Color)

Initializes a new instance of the LinearGradientBrush class.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public LinearGradientBrush (
    Point p1,
    Point p2,
    Color color1,
    Color color2
)
```

Visual Basic Copy Code

```
Public New ( _
    p1 As Point, _
```

```
    p2 As Point, _
    color1 As Color, _
    color2 As Color _
)
```

Parameters

p1

p2

color1

color2

See Also

LinearGradientBrush Constructor Overload List
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush Methods

LinearGradientBrush.Clone Method

Creates an exact copy of this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override Brush Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Brush
```

Return Value

An instance of the LinearGradientBrush class.

See Also

LinearGradientBrush Members
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush.Equals Method

Brush.Equals override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override bool Equals (
    Brush other
)
```

Visual Basic Copy Code

```
Public Overrides Function Equals( _
    other As Brush _
) As Boolean
```

Parameters

other
The Brush to compare with the current Brush.

Return Value

true if the specified Brush is equal to the current Brush; otherwise, false.

See Also

LinearGradientBrush Members
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush.GetHashCode Method

Brush.GetHashCode override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override int GetHashCode ()
```

Visual Basic Copy Code

```
Public Overrides Function GetHashCode () As Integer
```

Return Value

See Also

LinearGradientBrush Members
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush.ToColor Method

Brush.ToColor override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override Color ToColor ()
```

Visual Basic Copy Code

```
Public Overrides Function ToColor () As Color
```

Return Value

See Also

LinearGradientBrush Members
LinearGradientBrush Class

LinearGradientBrush Properties

LinearGradientBrush.Angle Property

Gets or sets the angle of the gradient's orientation line.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double Angle { get; set; }
```

Visual Basic Copy Code

```
Public Property Angle As Double
```

Property Value

A float value that represents he angle, measured in degrees clockwise from the x-axis, of the gradient's orientation line.

See Also

LinearGradientBrush Members
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush.Colors Property

Gets or sets an array of colors that represents the colors to use at corresponding positions along a gradient.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Color[] Colors { get; set; }
```

Visual Basic Copy Code

```
Public Property Colors As Color()
```

Property Value

See Also

LinearGradientBrush Members
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush.EndPoint Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Point EndPoint { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property EndPoint As Point
```

Property Value

See Also

LinearGradientBrush Members
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush.Positions Property

Gets or sets the positions along a gradient line.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double[] Positions { get; set; }
```

Visual Basic Copy Code

```
Public Property Positions As Double()
```

Property Value

See Also

LinearGradientBrush Members
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush.StartPoint Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Point StartPoint { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property StartPoint As Point
```

Property Value

See Also

LinearGradientBrush Members
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush.Transform Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IMatrix Transform { get; set; }
```

Visual Basic Copy Code

```
Public Property Transform As IMatrix
```

Property Value

See Also

LinearGradientBrush Members
LinearGradientBrush Class
MindFusion.Drawing Namespace

LinearGradientBrush.WrapMode Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public WrapMode WrapMode { get; set; }
```

Visual Basic Copy Code

```
Public Property WrapMode As WrapMode
```

Property Value

See Also

LinearGradientBrush Members
LinearGradientBrush Class
MindFusion.Drawing Namespace

LineJoin Enumeration

Describes the shape that joins two lines or segments.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum LineJoin
```

Visual Basic Copy Code

```
Public Enum LineJoin
```

Members

	Member name	Description
	Miter	Regular angular vertices.
	Bevel	Beveled vertices.
	Round	Rounded vertices.

See Also

MindFusion.Drawing Namespace

MatrixOrder Enumeration

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum MatrixOrder
```

Visual Basic Copy Code

```
Public Enum MatrixOrder
```

Members

	Member name	Description
	Prepend	
	Append	

See Also

MindFusion.Drawing Namespace

MauiGraphics Class

Syntax

C# Copy Code

```
public class MauiGraphics : IGraphics, IDisposable
```

Visual Basic Copy Code

```
Public Class MauiGraphics
    Implements IGraphics, IDisposable
```

Inheritance Hierarchy

System.Object
MindFusion.Drawing.MauiGraphics

See Also

MauiGraphics Members
MindFusion.Drawing Namespace

MauiGraphics Members

The following tables list the members exposed by the MauiGraphics type.

Public Constructors

	Name	Description
	MauiGraphics	Overloaded.

Public Properties

	Name	Description
	Clip	
	DpiX	
	DpiY	
	EnableAntialiasing	
	GraphicsType	
	SupportsPenWidth0	
	Transform	
	UnitScale	

Public Methods

	Name	Description
	BeginHyperLink	
	ClipImage	Overloaded.
	CreateImage	
	Dispose	
	DrawArc	Overloaded.
	DrawBezier	
	DrawBeziers	
	DrawCurve	
	DrawEllipse	Overloaded.
	DrawImage	Overloaded.
	DrawLine	Overloaded.
	DrawLines	

	DrawPath	
	DrawPoint	
	DrawPolygon	
	DrawRectangle	Overloaded.
	DrawString	Overloaded.
	EndHyperLink	
	FillEllipse	Overloaded.
	FillPath	
	FillPolygon	
	FillRectangle	Overloaded.
	GetFontHeight	
	GetImageSize	
	MeasureString	Overloaded.
	MultiplyTransform	
	PopOpacity	
	PushOpacity	
	ResetTransform	
	Restore	
	RotateTransform	
	Save	
	ScaleTransform	
	SetClip	Overloaded.
	SetTransform	
	TestDrawString	
	TranslateTransform	

See Also

MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics Constructor

Overload List

Name	Description
MauiGraphics (ICanvas)	
MauiGraphics (ICanvas, SkiaBitmapExportContextEx)	

See Also

MauiGraphics Constructor (ICanvas)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public MauiGraphics (
    ICanvas canvas
)
```

Visual Basic Copy Code

```
Public New ( _
    canvas As ICanvas _
)
```

Parameters

canvas

See Also

MauiGraphics Constructor Overload List
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics Constructor (ICanvas, SkiaBitmapExportContextEx)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public MauiGraphics (
    ICanvas canvas,
    SkiaBitmapExportContextEx skiaBitmapExportContext
)
```

Visual Basic Copy Code

```
Public New ( _
    canvas As ICanvas, _
    skiaBitmapExportContext As SkiaBitmapExportContextEx _
)
```

Parameters

canvas

skiaBitmapExportContext

See Also

MauiGraphics Constructor Overload List
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics Methods

MauiGraphics.BeginHyperLink Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void BeginHyperLink (
    string hyperlink,
    string target
)
```

Visual Basic Copy Code

```
Public Sub BeginHyperLink( _
    hyperlink As String, _
    target As String _
)
```

Parameters

hyperlink

target

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.ClipImage Method

Overload List

Name	Description
MauiGraphics.ClipImage (IImage, IGraphicsPath)	
MauiGraphics.ClipImage (IImage, Point[])	

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.ClipImage Method (IImage, IGraphicsPath)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IImage ClipImage (
    IImage sourceImage,
    IGraphicsPath clip
)
```

Visual Basic Copy Code

```
Public Function ClipImage( _
    sourceImage As IImage, _
    clip As IGraphicsPath _
) As IImage
```

Parameters

sourceImage

clip

Return Value

See Also

ClipImage Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.ClipImage Method (IImage, Point[])

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IImage ClipImage (
    IImage sourceImage,
    Point[] clipPoly
)
```

Visual Basic Copy Code

```
Public Function ClipImage( _
    sourceImage As IImage, _
    clipPoly() As Point _
) As IImage
```

Parameters

sourceImage

clipPoly

Return Value

See Also

ClipImage Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.CreateImage Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IImage CreateImage (
    IGraphicsPath clip = null
)
```

Visual Basic Copy Code

```
Public Function CreateImage( _
    Optional clip As IGraphicsPath = Nothing _
) As IImage
```

Parameters

clip
Optional.

Return Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.Dispose Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public virtual void Dispose ()
```

Visual Basic Copy Code

```
Public Overridable Sub Dispose ()
```

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawArc Method

Overload List

Name	Description
MauiGraphics.DrawArc (Pen, Double, Double, Double, Double, Double, Double)	
MauiGraphics.DrawArc (Pen, Rect, Double, Double)	

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawArc Method (Pen, Double, Double, Double, Double, Double, Double)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawArc (
    Pen pen,
    double x,
    double y,
    double width,
    double height,
    double startAngle,
    double sweepAngle
)
```

Visual Basic Copy Code

```
Public Sub DrawArc( _
    pen As Pen, _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double, _
    startAngle As Double, _
    sweepAngle As Double _
)
```

Parameters

pen

x

y

width

height

startAngle

sweepAngle

See Also

DrawArc Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawArc Method (Pen, Rect, Double, Double)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

Copy Code

```
public void DrawArc (
    Pen pen,
    Rect rect,
    double startAngle,
    double sweepAngle
)
```

Visual Basic

Copy Code

```
Public Sub DrawArc( _
    pen As Pen, _
    rect As Rect, _
    startAngle As Double, _
    sweepAngle As Double _
)
```

Parameters

pen

rect

startAngle

sweepAngle

See Also

DrawArc Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawBezier Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

Copy Code

```
public void DrawBezier (
    Pen pen,
    Point p1,
    Point p2,
    Point p3,
    Point p4
)
```

Visual Basic

Copy Code

```
Public Sub DrawBezier( _
    pen As Pen, _
    p1 As Point, _
    p2 As Point, _
    p3 As Point, _
    p4 As Point _
)
```

Parameters

pen

p1

p2

p3

p4

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawBeziers Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawBeziers (
    Pen pen,
    IList<Point> points
)
```

Visual Basic Copy Code

```
Public Sub DrawBeziers( _
    pen As Pen, _
    points As IList(Of Point) _
)
```

Parameters

pen

points

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawCurve Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawCurve (
    Pen pen,
    Point[] points
)
```

Visual Basic Copy Code

```
Public Sub DrawCurve( _
    pen As Pen, _
    points() As Point _
)
```

Parameters

pen

points

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawEllipse Method

Overload List

Name	Description
MauiGraphics.DrawEllipse (Pen, Double, Double, Double, Double)	
MauiGraphics.DrawEllipse (Pen, Rect)	

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawEllipse Method (Pen, Double, Double, Double, Double)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawEllipse (
    Pen pen,
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Public Sub DrawEllipse( _
    pen As Pen, _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
)
```

Parameters

pen

x

y

width

height

See Also

DrawEllipse Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawEllipse Method (Pen, Rect)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawEllipse (
    Pen pen,
    Rect rect
)
```

Visual Basic Copy Code

```
Public Sub DrawEllipse( _
    pen As Pen, _
    rect As Rect _
)
```

Parameters

pen

rect

See Also

DrawEllipse Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawImage Method

Overload List

Name	Description
MauiGraphics.DrawImage (IImage, Double, Double)	
MauiGraphics.DrawImage (IImage, Rect)	
MauiGraphics.DrawImage (IImage, Rect, ImageAlign, Double, Point)	

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawImage Method (IImage, Double, Double)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawImage (
    IImage image,
    double x,
    double y
)
```

Visual Basic Copy Code

```
Public Sub DrawImage( _
    image As IImage, _
    x As Double, _
    y As Double _
)
```

Parameters

image

x

y

See Also

DrawImage Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawImage Method (IImage, Rect)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawImage (
    IImage image,
    Rect rect
)
```

Visual Basic Copy Code

```
Public Sub DrawImage( _
    image As IImage, _
    rect As Rect _
)
```

Parameters

image

rect

See Also

DrawImage Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawImage Method (IImage, Rect, ImageAlign, Double, Point)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawImage (
    IImage image,
    Rect rect,
    ImageAlign imageAlign,
    double rotationAngle,
    Point pivot
)
```

Visual Basic Copy Code

```
Public Sub DrawImage( _
    image As IImage, _
    rect As Rect, _
    imageAlign As ImageAlign, _
    rotationAngle As Double, _
    pivot As Point _
)
```

Parameters

image

rect

imageAlign

rotationAngle

pivot

See Also

DrawImage Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawLine Method

Overload List

Name	Description
MauiGraphics.DrawLine (Pen, Double, Double, Double, Double)	
MauiGraphics.DrawLine (Pen, Point, Point)	

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawLine Method (Pen, Double, Double, Double, Double)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawLine (
    Pen pen,
    double x1,
    double y1,
    double x2,
    double y2
)
```

Visual Basic Copy Code

```
Public Sub DrawLine( _
    pen As Pen, _
    x1 As Double, _
    y1 As Double, _
    x2 As Double, _
    y2 As Double _
)
```

Parameters

pen

x1

y1

x2

y2

See Also

DrawLine Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawLine Method (Pen, Point, Point)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawLine (
    Pen pen,
    Point p1,
    Point p2
)
```

Visual Basic Copy Code

```
Public Sub DrawLine( _
    pen As Pen, _
    p1 As Point, _
```

```
    p2 As Point _
)
```

Parameters

pen

p1

p2

See Also

DrawLine Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawLines Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawLines (
    Pen pen,
    IList<Point> points
)
```

Visual Basic Copy Code

```
Public Sub DrawLines( _
    pen As Pen, _
    points As IList(Of Point) _
)
```

Parameters

pen

points

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawPath Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawPath (
    Pen pen,
    IGraphicsPath path
)
```

Visual Basic Copy Code

```
Public Sub DrawPath( _
    pen As Pen, _
    path As IGraphicsPath _
)
```

Parameters

pen

path

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawPoint Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawPoint (
    SolidBrush brush,
    double x,
    double y,
    double size
)
```

Visual Basic Copy Code

```
Public Sub DrawPoint( _
    brush As SolidBrush, _
    x As Double, _
    y As Double, _
    size As Double _
)
```

Parameters

brush

x

y

size

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawPolygon Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawPolygon (
    Pen pen,
    IList<Point> points
)
```

Visual Basic Copy Code

```
Public Sub DrawPolygon( _
    pen As Pen, _
    points As IList(Of Point) _
)
```

Parameters

pen

points

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawRectangle Method

Overload List

Name	Description
MauiGraphics.DrawRectangle (Pen, Double, Double, Double, Double)	
MauiGraphics.DrawRectangle (Pen, Rect)	

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawRectangle Method (Pen, Double, Double, Double, Double)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawRectangle (
    Pen pen,
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Public Sub DrawRectangle( _
    pen As Pen, _
    x As Double, _
```

```
    y As Double, _
    width As Double, _
    height As Double _
)
```

Parameters

pen

x

y

width

height

See Also

DrawRectangle Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawRectangle Method (Pen, Rect)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawRectangle (
    Pen pen,
    Rect rect
)
```

Visual Basic Copy Code

```
Public Sub DrawRectangle( _
    pen As Pen, _
    rect As Rect _
)
```

Parameters

pen

rect

See Also

DrawRectangle Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawString Method

Overload List

Name	Description
MauiGraphics.DrawString (String, Font, Brush, Double, Double)	

MauiGraphics.DrawString (String, Font, Brush, Point, StringFormat)	
MauiGraphics.DrawString (String, Font, Brush, Rect)	
MauiGraphics.DrawString (String, Font, Brush, Rect, StringFormat)	

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawString Method (String, Font, Brush, Double, Double)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawString (
    string text,
    Font font,
    Brush brush,
    double x,
    double y
)
```

Visual Basic Copy Code

```
Public Sub DrawString( _
    text As String, _
    font As Font, _
    brush As Brush, _
    x As Double, _
    y As Double _
)
```

Parameters

text

font

brush

x

y

See Also

DrawString Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawString Method (String, Font, Brush, Point, StringFormat)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawString (
    string text,
    Font font,
    Brush brush,
    Point location,
    StringFormat format
)
```

Visual Basic Copy Code

```
Public Sub DrawString( _
    text As String, _
    font As Font, _
    brush As Brush, _
    location As Point, _
    format As StringFormat _
)
```

Parameters

text

font

brush

location

format

See Also

DrawString Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawString Method (String, Font, Brush, Rect)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawString (
    string text,
    Font font,
    Brush brush,
    Rect rect
)
```

Visual Basic Copy Code

```
Public Sub DrawString( _
    text As String, _
    font As Font, _
    brush As Brush, _
    rect As Rect _
)
```

Parameters

text

font

brush

rect

See Also

DrawString Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DrawString Method (String, Font, Brush, Rect, StringFormat)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void DrawString (
    string text,
    Font font,
    Brush brush,
    Rect rect,
    StringFormat format
)
```

Visual Basic Copy Code

```
Public Sub DrawString( _
    text As String, _
    font As Font, _
    brush As Brush, _
    rect As Rect, _
    format As StringFormat _
)
```

Parameters

text

font

brush

rect

format

See Also

DrawString Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.EndHyperLink Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void EndHyperLink ()
```

Visual Basic Copy Code

```
Public Sub EndHyperLink ()
```

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.FillEllipse Method

Overload List

Name	Description
MauiGraphics.FillEllipse (Brush, Double, Double, Double, Double)	
MauiGraphics.FillEllipse (Brush, Rect)	

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.FillEllipse Method (Brush, Double, Double, Double, Double)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void FillEllipse (
    Brush brush,
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Public Sub FillEllipse( _
    brush As Brush, _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
)
```

Parameters

brush

x

y

width

height

See Also

FillEllipse Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.FillEllipse Method (Brush, Rect)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void FillEllipse (
    Brush brush,
    Rect rect
)
```

Visual Basic Copy Code

```
Public Sub FillEllipse( _
    brush As Brush, _
    rect As Rect _
)
```

Parameters

brush

rect

See Also

FillEllipse Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.FillPath Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void FillPath (
    Brush brush,
    IGraphicsPath path
)
```

Visual Basic Copy Code

```
Public Sub FillPath( _
    brush As Brush, _
```

```
    path As IGraphicsPath _
)
```

Parameters

brush

path

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.FillPolygon Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void FillPolygon (
    Brush brush,
    IList<Point> points
)
```

Visual Basic Copy Code

```
Public Sub FillPolygon( _
    brush As Brush, _
    points As IList(Of Point) _
)
```

Parameters

brush

points

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.FillRectangle Method

Overload List

Name	Description
MauiGraphics.FillRectangle (Brush, Double, Double, Double, Double)	
MauiGraphics.FillRectangle (Brush, Rect)	

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.FillRectangle Method (Brush, Double, Double, Double, Double)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void FillRectangle (
    Brush brush,
    double x,
    double y,
    double width,
    double height
)
```

Visual Basic Copy Code

```
Public Sub FillRectangle( _
    brush As Brush, _
    x As Double, _
    y As Double, _
    width As Double, _
    height As Double _
)
```

Parameters

brush

x

y

width

height

See Also

FillRectangle Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.FillRectangle Method (Brush, Rect)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void FillRectangle (
    Brush brush,
    Rect rect
)
```

Visual Basic Copy Code

```
Public Sub FillRectangle( _
    brush As Brush, _
    rect As Rect _
)
```

Parameters

brush

rect

See Also

FillRectangle Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.GetFontHeight Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double GetFontHeight (
    Font font
)
```

Visual Basic Copy Code

```
Public Function GetFontHeight( _
    font As Font _
) As Double
```

Parameters

font

Return Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.GetImageSize Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Size GetImageSize (
    IImage image
)
```

Visual Basic Copy Code

```
Public Function GetImageSize( _
    image As IImage _
) As Size
```

Parameters

image

Return Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.MeasureString Method

Overload List

Name	Description
MauiGraphics.MeasureString (String, Font)	
MauiGraphics.MeasureString (String, Font, Double, StringFormat)	
MauiGraphics.MeasureString (String, Font, Size, StringFormat, Int32@, Int32@)	

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.MeasureString Method (String, Font)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Size MeasureString (
    string text,
    Font font
)
```

Visual Basic Copy Code

```
Public Function MeasureString( _
    text As String, _
    font As Font _
) As Size
```

Parameters

text

font

Return Value

See Also

MeasureString Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.MeasureString Method (String, Font, Double, StringFormat)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Size MeasureString (
    string text,
    Font font,
    double maxWidth,
    StringFormat format
)
```

Visual Basic Copy Code

```
Public Function MeasureString( _
    text As String, _
    font As Font, _
    maxWidth As Double, _
    format As StringFormat _
) As Size
```

Parameters

text

font

maxWidth

format

Return Value

See Also

MeasureString Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.MeasureString Method (String, Font, Size, StringFormat, Int32@, Int32@)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Size MeasureString (
    string text,
    Font font,
    Size layoutArea,
    StringFormat stringFormat,
    out int charactersFitted,
    out int linesFilled
)
```

Visual Basic Copy Code

```
Public Function MeasureString( _
    text As String, _
    font As Font, _
    layoutArea As Size, _
    stringFormat As StringFormat, _
    <System.Runtime.InteropServices.Out()> ByRef charactersFitted As Integer, _
    <System.Runtime.InteropServices.Out()> ByRef linesFilled As Integer _
) As Size
```

Parameters

text

font

layoutArea

stringFormat

charactersFitted

linesFilled

Return Value

See Also

MeasureString Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.MultiplyTransform Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void MultiplyTransform (
    IMatrix m,
    MatrixOrder order
)
```

Visual Basic Copy Code

```
Public Sub MultiplyTransform( _
    m As IMatrix, _
    order As MatrixOrder _
)
```

Parameters

m

order

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.PopOpacity Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void PopOpacity ()
```

Visual Basic Copy Code

```
Public Sub PopOpacity ()
```

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.PushOpacity Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void PushOpacity (
    double opacity
)
```

Visual Basic Copy Code

```
Public Sub PushOpacity( _
    opacity As Double _
)
```

Parameters

opacity

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.ResetTransform Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void ResetTransform ()
```

Visual Basic Copy Code

```
Public Sub ResetTransform ()
```

See Also

MauiGraphics.Restore Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void Restore (
    Object state
)
```

Visual Basic Copy Code

```
Public Sub Restore( _
    state As Object _
)
```

Parameters

state

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.RotateTransform Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void RotateTransform (
    double angle
)
```

Visual Basic Copy Code

```
Public Sub RotateTransform( _
    angle As Double _
)
```

Parameters

angle

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.Save Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Object Save ()
```

Visual Basic Copy Code

```
Public Function Save () As Object
```

Return Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.ScaleTransform Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void ScaleTransform (
    double sx,
    double sy
)
```

Visual Basic Copy Code

```
Public Sub ScaleTransform( _
    sx As Double, _
    sy As Double _
)
```

Parameters

sx

sy

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.SetClip Method

Overload List

Name	Description

MauiGraphics.SetClip (IGraphicsPath)	
MauiGraphics.SetClip (IGraphicsPath, CombineMode)	
MauiGraphics.SetClip (Rect)	
MauiGraphics.SetClip (Rect, CombineMode)	

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.SetClip Method (IGraphicsPath)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void SetClip (
    IGraphicsPath path
)
```

Visual Basic Copy Code

```
Public Sub SetClip( _
    path As IGraphicsPath _
)
```

Parameters

path

See Also

SetClip Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.SetClip Method (IGraphicsPath, CombineMode)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void SetClip (
    IGraphicsPath path,
    CombineMode mode
)
```

Visual Basic Copy Code

```
Public Sub SetClip( _
    path As IGraphicsPath, _
    mode As CombineMode _
)
```

Parameters

path

mode

See Also

SetClip Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.SetClip Method (Rect)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void SetClip (
    Rect rect
)
```

Visual Basic Copy Code

```
Public Sub SetClip( _
    rect As Rect _
)
```

Parameters

rect

See Also

SetClip Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.SetClip Method (Rect, CombineMode)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void SetClip (
    Rect rect,
    CombineMode mode
)
```

Visual Basic Copy Code

```
Public Sub SetClip( _
    rect As Rect, _
```

```
    mode As CombineMode _
)
```

Parameters

rect

mode

See Also

SetClip Method Overload List
MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.SetTransform Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void SetTransform (
    IMatrix m
)
```

Visual Basic Copy Code

```
Public Sub SetTransform( _
    m As IMatrix _
)
```

Parameters

m

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.TestDrawString Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void TestDrawString ()
```

Visual Basic Copy Code

```
Public Sub TestDrawString ()
```

See Also

MauiGraphics.TranslateTransform Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void TranslateTransform (
    double dx,
    double dy
)
```

Visual Basic Copy Code

```
Public Sub TranslateTransform( _
    dx As Double, _
    dy As Double _
)
```

Parameters

dx

dy

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics Properties

MauiGraphics.Clip Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IGraphicsPath Clip { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Clip As IGraphicsPath
```

Property Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DpiX Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double DpiX { get; set; }
```

Visual Basic Copy Code

```
Public Property DpiX As Double
```

Property Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.DpiY Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double DpiY { get; set; }
```

Visual Basic Copy Code

```
Public Property DpiY As Double
```

Property Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.EnableAntialiasing Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public bool EnableAntialiasing { get; set; }
```

Visual Basic Copy Code

```
Public Property EnableAntialiasing As Boolean
```

Property Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.GraphicsType Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public GraphicsType GraphicsType { get; set; }
```

Visual Basic Copy Code

```
Public Property GraphicsType As GraphicsType
```

Property Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.SupportsPenWidth0 Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public bool SupportsPenWidth0 { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property SupportsPenWidth0 As Boolean
```

Property Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.Transform Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IMatrix Transform { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property Transform As IMatrix
```

Property Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiGraphics.UnitScale Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double UnitScale { get; set; }
```

Visual Basic Copy Code

```
Public Property UnitScale As Double
```

Property Value

See Also

MauiGraphics Members
MauiGraphics Class
MindFusion.Drawing Namespace

MauiMatrix Class

Syntax

C# Copy Code

```
public sealed class MauiMatrix : IMatrix, IDisposable
```

Visual Basic Copy Code

```
Public NotInheritable Class MauiMatrix
    Implements IMatrix, IDisposable
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.MauiMatrix

See Also

MauiMatrix Members

The following tables list the members exposed by the MauiMatrix type.

Public Constructors

	Name	Description
	MauiMatrix	Overloaded.

Public Properties

	Name	Description
	Elements	
	IsIdentity	
	OffsetX	
	OffsetY	

Public Methods

	Name	Description
	Clone	
	Dispose	
	GetMatrix3x2	
	Inverse	
	Multiply	
	Reset	
	RotateAt	Overloaded.
	Scale	
	Shear	
	TransformPoint	
	TransformPoints	
	TransformRect	
	Translate	

See Also

MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix Constructor

Overload List

Name	Description
MauiMatrix ()	
MauiMatrix (Matrix3x2)	
MauiMatrix (Single, Single, Single, Single, Single, Single)	

See Also

MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix Constructor ()

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public MauiMatrix ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

MauiMatrix Constructor Overload List
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix Constructor (Matrix3x2)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public MauiMatrix (
    Matrix3x2 matrix
)
```

Visual Basic Copy Code

```
Public New ( _
    matrix As Matrix3x2 _
)
```

Parameters

matrix

See Also

MauiMatrix Constructor Overload List
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix Constructor (Single, Single, Single, Single, Single, Single)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public MauiMatrix (
    float m11,
    float m12,
    float m21,
    float m22,
    float m31,
    float m32
)
```

Visual Basic Copy Code

```
Public New ( _
    m11 As Single, _
    m12 As Single, _
    m21 As Single, _
    m22 As Single, _
    m31 As Single, _
    m32 As Single _
)
```

Parameters

m11

m12

m21

m22

m31

m32

See Also

MauiMatrix Constructor Overload List
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix Methods

MauiMatrix.Clone Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IMatrix Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As IMatrix
```

Return Value

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.Dispose Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void Dispose ()
```

Visual Basic Copy Code

```
Public Sub Dispose ()
```

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.GetMatrix3x2 Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Matrix3x2 GetMatrix3x2 ()
```

Visual Basic Copy Code

```
Public Function GetMatrix3x2 () As Matrix3x2
```

Return Value

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.Inverse Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
public IMatrix Inverse ()
```

Visual Basic

```
Public Function Inverse () As IMatrix
```

Return Value

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.Multiply Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
public void Multiply (
    IMatrix m,
    MatrixOrder order
)
```

Visual Basic

```
Public Sub Multiply( _
    m As IMatrix, _
    order As MatrixOrder _
)
```

Parameters

m

order

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.Reset Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
public void Reset ()
```

Visual Basic

```
Public Sub Reset ()
```

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.RotateAt Method

Overload List

Name	Description
MauiMatrix.RotateAt (Double)	
MauiMatrix.RotateAt (Double, Point)	

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.RotateAt Method (Double)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void RotateAt (
    double angle
)
```

Visual Basic Copy Code

```
Public Sub RotateAt( _
    angle As Double _
)
```

Parameters

angle

See Also

RotateAt Method Overload List
MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.RotateAt Method (Double, Point)

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void RotateAt (
    double angle,
    Point point
)
```

Visual Basic Copy Code

```
Public Sub RotateAt( _
    angle As Double, _
    point As Point _
)
```

Parameters

angle

point

See Also

RotateAt Method Overload List
MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.Scale Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void Scale (
    double sx,
    double sy
)
```

Visual Basic Copy Code

```
Public Sub Scale( _
    sx As Double, _
    sy As Double _
)
```

Parameters

sx

sy

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.Shear Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void Shear (
    double sx,
    double sy
)
```

Visual Basic Copy Code

```
Public Sub Shear( _
    sx As Double, _
    sy As Double _
)
```

Parameters

sx

sy

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.TransformPoint Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Point TransformPoint (
    Point point
)
```

Visual Basic Copy Code

```
Public Function TransformPoint( _
    point As Point _
) As Point
```

Parameters

point

Return Value

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.TransformPoints Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void TransformPoints (
    IList<Point> points
)
```

Visual Basic Copy Code

```
Public Sub TransformPoints( _
    points As IList(Of Point) _
)
```

Parameters

points

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.TransformRect Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Rect TransformRect (
    Rect rect
)
```

Visual Basic Copy Code

```
Public Function TransformRect( _
    rect As Rect _
) As Rect
```

Parameters

rect

Return Value

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.Translate Method

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public void Translate (
    double dx,
    double dy
)
```

Visual Basic

```
Public Sub Translate( _
    dx As Double, _
    dy As Double _
)
```

Parameters

dx

dy

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix Properties

MauiMatrix.Elements Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
public float[] Elements { get; }
```

Visual Basic

```
Public ReadOnly Property Elements As Single()
```

Property Value

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.IsIdentity Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C#

```
public bool IsIdentity { get; }
```

Visual Basic

```
Public ReadOnly Property IsIdentity As Boolean
```

Property Value

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.OffsetX Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double OffsetX { get; set; }
```

Visual Basic Copy Code

```
Public Property OffsetX As Double
```

Property Value

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

MauiMatrix.OffsetY Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double OffsetY { get; set; }
```

Visual Basic Copy Code

```
Public Property OffsetY As Double
```

Property Value

See Also

MauiMatrix Members
MauiMatrix Class
MindFusion.Drawing Namespace

PathGradientBrush Class

PathGradientBrush represents a brush that fills the interiors of geometric shapes with a gradient.

Syntax

C# Copy Code

```
public class PathGradientBrush : Brush
```

Visual Basic Copy Code

```
Public Class PathGradientBrush
    Inherits Brush
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.Brush
    MindFusion.Drawing.PathGradientBrush

See Also

PathGradientBrush Members
MindFusion.Drawing Namespace

PathGradientBrush Members

PathGradientBrush represents a brush that fills the interiors of geometric shapes with a gradient.

The following tables list the members exposed by the **PathGradientBrush** type.

Public Constructors

	Name	Description
	PathGradientBrush	Initializes a new instance of the PathGradientBrush class.

Public Properties

	Name	Description
	Colors	Gets or sets an array of colors that represents the colors to use at corresponding positions along a gradient.
	Path	Gets or sets the IGraphicsPath that defines the area filled by this brush.
	Positions	Gets or sets the positions along a gradient line.
	Transform	
	WrapMode	

Public Methods

	Name	Description
	Clone	Creates an exact copy of this brush.
	Equals	Brush.Equals override.
	GetHashCode	Brush.GetHashCode override.
	ToColor	Brush.ToColor override.

See Also

PathGradientBrush Class
MindFusion.Drawing Namespace

PathGradientBrush Constructor

Initializes a new instance of the PathGradientBrush class.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public PathGradientBrush (
    IGraphicsPath path
)
```

Visual Basic Copy Code

```
Public New ( _
    path As IGraphicsPath _
)
```

Parameters

path
The IGraphicsPath that defines the area filled by this brush.

See Also

PathGradientBrush Class
MindFusion.Drawing Namespace

PathGradientBrush Methods

PathGradientBrush.Clone Method

Creates an exact copy of this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override Brush Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Brush
```

Return Value

An instance of the PathGradientBrush class.

See Also

PathGradientBrush Members
PathGradientBrush Class
MindFusion.Drawing Namespace

PathGradientBrush.Equals Method

Brush.Equals override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override bool Equals (
    Brush other
)
```

Visual Basic Copy Code

```
Public Overrides Function Equals( _
    other As Brush _
) As Boolean
```

Parameters

other

The Brush to compare with the current Brush.

Return Value

true if the specified Brush is equal to the current Brush; otherwise, false.

See Also

PathGradientBrush Members
PathGradientBrush Class
MindFusion.Drawing Namespace

PathGradientBrush.GetHashCode Method

Brush.GetHashCode override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override int GetHashCode ()
```

Visual Basic Copy Code

```
Public Overrides Function GetHashCode () As Integer
```

Return Value

See Also

PathGradientBrush Members
PathGradientBrush Class
MindFusion.Drawing Namespace

PathGradientBrush.ToColor Method

Brush.ToColor override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override Color ToColor ()
```

Visual Basic Copy Code

```
Public Overrides Function ToColor () As Color
```

Return Value

See Also

PathGradientBrush Members
PathGradientBrush Class
MindFusion.Drawing Namespace

PathGradientBrush Properties

PathGradientBrush.Colors Property

Gets or sets an array of colors that represents the colors to use at corresponding positions along a gradient.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Color[] Colors { get; set; }
```

Visual Basic Copy Code

```
Public Property Colors As Color()
```

Property Value

See Also

PathGradientBrush Members
PathGradientBrush Class
MindFusion.Drawing Namespace

PathGradientBrush.Path Property

Gets or sets the IGraphicsPath that defines the area filled by this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IGraphicsPath Path { get; set; }
```

Visual Basic Copy Code

```
Public Property Path As IGraphicsPath
```

Property Value

See Also

PathGradientBrush Members
PathGradientBrush Class
MindFusion.Drawing Namespace

PathGradientBrush.Positions Property

Gets or sets the positions along a gradient line.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double[] Positions { get; set; }
```

Visual Basic Copy Code

```
Public Property Positions As Double()
```

Property Value

See Also

PathGradientBrush Members
PathGradientBrush Class
MindFusion.Drawing Namespace

PathGradientBrush.Transform Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IMatrix Transform { get; set; }
```

Visual Basic Copy Code

```
Public Property Transform As IMatrix
```

Property Value

See Also

PathGradientBrush Members
PathGradientBrush Class
MindFusion.Drawing Namespace

PathGradientBrush.WrapMode Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public WrapMode WrapMode { get; set; }
```

Visual Basic Copy Code

```
Public Property WrapMode As WrapMode
```

Property Value

See Also

PathGradientBrush Members
PathGradientBrush Class
MindFusion.Drawing Namespace

Pen Class

Pen objects are used to draw the outline of geometric shapes.

Syntax

C# Copy Code

```
public class Pen
```

Visual Basic Copy Code

```
Public Class Pen
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.Pen

See Also

Pen Members
MindFusion.Drawing Namespace

Pen Members

Pen objects are used to draw the outline of geometric shapes.

The following tables list the members exposed by the Pen type.

Public Constructors

	Name	Description
	Pen	Overloaded. Initialize a new instance of the **Pen** class.

Public Properties

	Name	Description
	Alignment	
	Brush	Gets or sets the Brush object that determines attributes of this pen.
	Color	Gets or sets the color of this pen.
	DashOffset	Gets or sets the distance from the start of a line to the beginning of a dash pattern.
	DashPattern	Gets or sets an array of custom dashes and spaces.
	DashStyle	Gets or sets the style used for dashed lines drawn with this pen.
	EndCap	
	LineJoin	Gets or sets the join style for the ends of two consecutive lines drawn with this pen.

	MiterLimit	Gets or sets the limit of the thickness of the join on a mitered corner.
	StartCap	
	Transform	
	Unit	Gets or sets the unit of the pen width.
	Width	Gets or sets the width of the pen.

Public Methods

	Name	Description
	Clone	Creates an exact copy of this pen.

See Also

Pen Class
MindFusion.Drawing Namespace

Pen Constructor

Initialize a new instance of the **Pen** class.

Overload List

Name	Description
Pen (Brush, Double)	Initializes a new instance of the Pen class with the specified Brush object and the specified width.
Pen (Color)	Initializes a new instance of the **Pen** class with the specified color.
Pen (Color, Double)	Initializes a new instance of the **Pen** class with the specified color and width.
Pen (Color, Double, PenUnit)	Initializes a new instance of the **Pen** class.

See Also

Pen Class
MindFusion.Drawing Namespace

Pen Constructor (Brush, Double)

Initializes a new instance of the Pen class with the specified Brush object and the specified width.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Pen (
    Brush brush,
    double width
)
```

Visual Basic Copy Code

```
Public New ( _
    brush As Brush, _
```

```
    width As Double _
)
```

Parameters

brush

A Brush instance that determines the fill properties of the new pen.

width

The width of the new pen.

See Also

Pen Constructor Overload List
Pen Class
MindFusion.Drawing Namespace

Pen Constructor (Color)

Initializes a new instance of the Pen class with the specified color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Pen (
    Color color
)
```

Visual Basic Copy Code

```
Public New ( _
    color As Color _
)
```

Parameters

color

A Color structure that specifies the color of the new pen.

See Also

Pen Constructor Overload List
Pen Class
MindFusion.Drawing Namespace

Pen Constructor (Color, Double)

Initializes a new instance of the Pen class with the specified color and width.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Pen (
    Color color,
    double width
)
```

Visual Basic Copy Code

```
Public New ( _
    color As Color, _
```

```
    width As Double _
)
```

Parameters

color

A Color structure that specifies the color of the new pen.

width

The width of the new pen.

See Also

Pen Constructor Overload List
Pen Class
MindFusion.Drawing Namespace

Pen Constructor (Color, Double, PenUnit)

Initializes a new instance of the Pen class.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Pen (
    Color color,
    double width,
    PenUnit unit
)
```

Visual Basic Copy Code

```
Public New ( _
    color As Color, _
    width As Double, _
    unit As PenUnit _
)
```

Parameters

color

The initial color of the pen.

width

The initial width of the pen.

unit

The initial unit of the pen width.

See Also

Pen Constructor Overload List
Pen Class
MindFusion.Drawing Namespace

Pen Methods

Pen.Clone Method

Creates an exact copy of this pen.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Pen Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As Pen
```

Return Value

An instance of the **Pen** class.

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen Properties

Pen.Alignment Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public PenAlignment Alignment { get; set; }
```

Visual Basic Copy Code

```
Public Property Alignment As PenAlignment
```

Property Value

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.Brush Property

Gets or sets the Brush object that determines attributes of this pen.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Brush Brush { get; set; }
```

Visual Basic Copy Code

```
Public Property Brush As Brush
```

Property Value

An instance of the Brush class that determines attributes of this pen.

Remarks

Setting this property causes the pen to draw filled lines and curves. It overrides the Color property of the pen.

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.Color Property

Gets or sets the color of this pen.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Color Color { get; set; }
```

Visual Basic Copy Code

```
Public Property Color As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** that represents the color of this pen.

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.DashOffset Property

Gets or sets the distance from the start of a line to the beginning of a dash pattern.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public float DashOffset { get; set; }
```

Visual Basic Copy Code

```
Public Property DashOffset As Single
```

Property Value

A float value representing the distance from the start of a line to the beginning of a dash pattern.

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.DashPattern Property

Gets or sets an array of custom dashes and spaces.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double[] DashPattern { get; set; }
```

Visual Basic Copy Code

```
Public Property DashPattern As Double()
```

Property Value

An array of real numbers that specify the lengths of alternating dashes and spaces in dashed lines.

Remarks

The elements in the array set the length of each dash and space in the dash pattern. The first element sets the length of a dash, the second element sets the length of a space, the third element sets the length of a dash, and so on.The length of each dash and space in the dash pattern is the product of the element value in the array and the width of the pen.

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.DashStyle Property

Gets or sets the style used for dashed lines drawn with this pen.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public DashStyle DashStyle { get; set; }
```

Visual Basic Copy Code

```
Public Property DashStyle As DashStyle
```

Property Value

A member of the DashStyle enumeration that represents the style used for dashed lines.

Remarks

A value of **DashStyle.Custom** for this property specifies that a custom pattern of dashes and spaces, defined by the DashPattern property, make up lines drawn with this pen.

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.EndCap Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public LineCap EndCap { get; set; }
```

Visual Basic Copy Code

```
Public Property EndCap As LineCap
```

Property Value

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.LineJoin Property

Gets or sets the join style for the ends of two consecutive lines drawn with this pen.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public LineJoin LineJoin { get; set; }
```

Visual Basic Copy Code

```
Public Property LineJoin As LineJoin
```

Property Value

A member of the LineJoin enumeration that represents the join style for the ends of two consecutive lines.

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.MiterLimit Property

Gets or sets the limit of the thickness of the join on a mitered corner.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double MiterLimit { get; set; }
```

Visual Basic Copy Code

```
Public Property MiterLimit As Double
```

Property Value

A floag value representing the limit of the thickness of the join on a mitered corner. The default is 10.0f.

Remarks

The miter length is the distance from the intersection of the line walls on the inside of the join to the intersection of the line walls outside of the join. The miter length can be large when the angle between two lines is small. The miter limit is the maximum allowed ratio of miter length to stroke width. If the miter length of the join of the intersection exceeds the limit of the join, then the join will be beveled to keep it within the limit of the join of the intersection.

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.StartCap Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public LineCap StartCap { get; set; }
```

Visual Basic Copy Code

```
Public Property StartCap As LineCap
```

Property Value

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.Transform Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IMatrix Transform { get; set; }
```

Visual Basic Copy Code

```
Public Property Transform As IMatrix
```

Property Value

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.Unit Property

Gets or sets the unit of the pen width.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public PenUnit Unit { get; set; }
```

Visual Basic Copy Code

```
Public Property Unit As PenUnit
```

Property Value

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

Pen.Width Property

Gets or sets the width of the pen.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public double Width { get; set; }
```

Visual Basic Copy Code

```
Public Property Width As Double
```

Property Value

A float value indicating the width of the pen.

See Also

Pen Members
Pen Class
MindFusion.Drawing Namespace

PenAlignment Enumeration

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum PenAlignment
```

Visual Basic Copy Code

```
Public Enum PenAlignment
```

Members

	Member name	Description
	Inset	

See Also

MindFusion.Drawing Namespace

PenUnit Enumeration

Specifies the unit of the width of a pen.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum PenUnit
```

Visual Basic Copy Code

```
Public Enum PenUnit
```

Members

	Member name	Description
	Pixel	Indicates that the width of the pen is expressed as physical pixels.
	Dip	Indicates that the width of the pen is expressed as device-independent pixels.

See Also

MindFusion.Drawing Namespace

SolidBrush Class

SolidBrush objects represent brushes of a single color that are used to fill the interiors of geometric shapes.

Syntax

C# Copy Code

```
public class SolidBrush : Brush
```

Visual Basic Copy Code

```
Public Class SolidBrush
    Inherits Brush
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.Brush
    MindFusion.Drawing.SolidBrush

See Also

SolidBrush Members
MindFusion.Drawing Namespace

SolidBrush Members

SolidBrush objects represent brushes of a single color that are used to fill the interiors of geometric shapes.

The following tables list the members exposed by the **SolidBrush** type.

Public Constructors

	Name	Description
	SolidBrush	Initializes a new instance of the **SolidBrush** class with the specified color.

Public Properties

	Name	Description
	Color	Gets or sets the color of the brush.

Public Methods

	Name	Description
	Clone	Creates an exact copy of this brush.
	Equals	Brush.Equals override.
	GetHashCode	Brush.GetHashCode override.
	ToColor	Brush.ToColor override.

See Also

SolidBrush Class
MindFusion.Drawing Namespace

SolidBrush Constructor

Initializes a new instance of the SolidBrush class with the specified color.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public SolidBrush (
    Color color
)
```

Visual Basic Copy Code

```
Public New ( _
    color As Color _
)
```

Parameters

color
A **Color** object that specifies the color of the new brush.

See Also

SolidBrush Class
MindFusion.Drawing Namespace

SolidBrush Methods

SolidBrush.Clone Method

Creates an exact copy of this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override Brush Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Brush
```

Return Value

An instance of the SolidBrush class.

See Also

SolidBrush Members
SolidBrush Class
MindFusion.Drawing Namespace

SolidBrush.Equals Method

Brush.Equals override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override bool Equals (
    Brush other
)
```

Visual Basic Copy Code

```
Public Overrides Function Equals( _
    other As Brush _
) As Boolean
```

Parameters

other
The Brush to compare with the current Brush.

Return Value

true if the specified Brush is equal to the current Brush; otherwise, false.

See Also

SolidBrush.GetHashCode Method

Brush.GetHashCode override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override int GetHashCode ()
```

Visual Basic Copy Code

```
Public Overrides Function GetHashCode () As Integer
```

Return Value

See Also

SolidBrush Members
SolidBrush Class
MindFusion.Drawing Namespace

SolidBrush.ToColor Method

Brush.ToColor override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override Color ToColor ()
```

Visual Basic Copy Code

```
Public Overrides Function ToColor () As Color
```

Return Value

See Also

SolidBrush Members
SolidBrush Class
MindFusion.Drawing Namespace

SolidBrush Properties

SolidBrush.Color Property

Gets or sets the color of the brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public Color Color { get; set; }
```

Visual Basic Copy Code

```
Public Property Color As Color
```

Property Value

A **Microsoft.Maui.Graphics.Color** that represents the color of this brush.

See Also

SolidBrush Members
SolidBrush Class
MindFusion.Drawing Namespace

StringAlignment Enumeration

Specifies the alignment of text relative to a point or inside a rectangle.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum StringAlignment
```

Visual Basic Copy Code

```
Public Enum StringAlignment
```

Members

	Member name	Description
	Near	Left side of text is aligned to target point, or to the left side of the layout rectangle.
	Center	Center of text is aligned to target point, or to the center of the layout rectangle.
	Far	Right side of text is aligned to target point, or to the Right side of the layout rectangle.

See Also

MindFusion.Drawing Namespace

StringFormat Class

Encapsulates text layout information (such as alignment).

Syntax

C# Copy Code

```
public class StringFormat
```

Visual Basic	Copy Code
`Public Class StringFormat`	

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.StringFormat

See Also

StringFormat Members
MindFusion.Drawing Namespace

StringFormat Members

Encapsulates text layout information (such as alignment).

The following tables list the members exposed by the StringFormat type.

Public Constructors

	Name	Description
	StringFormat	Overloaded.

Public Fields

	Name	Description
	Default	The default string format.

Public Properties

	Name	Description
	Alignment	Gets or sets horizontal alignment of the string.
	ClipText	Gets or sets a value indicating whether to clip the text to bounds.
	LineAlignment	Gets or sets the vertical alignment of the string.
	WrapText	Gets or sets a value indicating whether to wrap the text.

Public Methods

	Name	Description
	Clone	Creates an exact clone of this object.
	Equals	Determines whether the specified object is equal to the current StringFormat.
	GetHashCode	Serves as a hash function for a StringFormat object.

See Also

StringFormat Class
MindFusion.Drawing Namespace

StringFormat Fields

StringFormat.Default Field

The default string format.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public static readonly StringFormat Default
```

Visual Basic Copy Code

```
Public Shared ReadOnly Default As StringFormat
```

See Also

StringFormat Members
StringFormat Class
MindFusion.Drawing Namespace

StringFormat Constructor

Overload List

Name	Description
StringFormat ()	Initializes a new instance of the StringFormat class.
StringFormat (StringFormat)	Initializes a new instance of the StringFormat class.

See Also

StringFormat Class
MindFusion.Drawing Namespace

StringFormat Constructor ()

Initializes a new instance of the StringFormat class.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public StringFormat ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

StringFormat Constructor Overload List
StringFormat Class
MindFusion.Drawing Namespace

StringFormat Constructor (StringFormat)

Initializes a new instance of the StringFormat class.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public StringFormat (
    StringFormat format
)
```

Visual Basic Copy Code

```
Public New ( _
    format As StringFormat _
)
```

Parameters

format
The StringFormat object from which to initialize this object.

See Also

StringFormat Constructor Overload List
StringFormat Class
MindFusion.Drawing Namespace

StringFormat Methods

StringFormat.Clone Method

Creates an exact clone of this object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public StringFormat Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As StringFormat
```

Return Value

An instance of the StringFormat class.

See Also

StringFormat Members
StringFormat Class
MindFusion.Drawing Namespace

StringFormat.Equals Method

Determines whether the specified object is equal to the current StringFormat.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override bool Equals (
    Object obj
)
```

Visual Basic Copy Code

```
Public Overrides Function Equals( _
    obj As Object _
) As Boolean
```

Parameters

obj
The object to compare with the current StringFormat.

Return Value

true if the specified object is equal to the current StringFormat; otherwise, false.

See Also

StringFormat Members
StringFormat Class
MindFusion.Drawing Namespace

StringFormat.GetHashCode Method

Serves as a hash function for a StringFormat object.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override int GetHashCode ()
```

Visual Basic Copy Code

```
Public Overrides Function GetHashCode () As Integer
```

Return Value

A hash code for this instance that is suitable for use in hashing algorithms and data structures such as a hash table.

See Also

StringFormat Members
StringFormat Class
MindFusion.Drawing Namespace

StringFormat Properties

StringFormat.Alignment Property

Gets or sets horizontal alignment of the string.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public StringAlignment Alignment { get; set; }
```

Visual Basic Copy Code

```
Public Property Alignment As StringAlignment
```

Property Value

See Also

StringFormat Members
StringFormat Class
MindFusion.Drawing Namespace

StringFormat.ClipText Property

Gets or sets a value indicating whether to clip the text to bounds.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public bool ClipText { get; set; }
```

Visual Basic Copy Code

```
Public Property ClipText As Boolean
```

Property Value

See Also

StringFormat Members
StringFormat Class
MindFusion.Drawing Namespace

StringFormat.LineAlignment Property

Gets or sets the vertical alignment of the string.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public StringAlignment LineAlignment { get; set; }
```

Visual Basic Copy Code

```
Public Property LineAlignment As StringAlignment
```

Property Value

See Also

StringFormat Members
StringFormat Class
MindFusion.Drawing Namespace

StringFormat.WrapText Property

Gets or sets a value indicating whether to wrap the text.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public bool WrapText { get; set; }
```

Visual Basic Copy Code

```
Public Property WrapText As Boolean
```

Property Value

See Also

StringFormat Members
StringFormat Class
MindFusion.Drawing Namespace

TextAlignment Enumeration

Specifies the alignment of text on the drawing surface.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum TextAlignment
```

Visual Basic Copy Code

```
Public Enum TextAlignment
```

Members

	Member name	Description
	Unspecified	No alignment is specified.
	TopLeft	Text is vertically aligned at the top, and horizontally aligned on the left.
	TopCenter	Text is vertically aligned at the top, and horizontally aligned at the center.
	TopRight	Text is vertically aligned at the top, and horizontally aligned on the right.
	MiddleLeft	Text is vertically aligned in the middle, and horizontally aligned on the left.
	MiddleCenter	Text is vertically aligned in the middle, and horizontally aligned at the center.
	MiddleRight	Text is vertically aligned in the middle, and horizontally aligned on the right.

	BottomLeft	Text is vertically aligned at the bottom, and horizontally aligned on the left.
	BottomCenter	Text is vertically aligned at the bottom, and horizontally aligned at the center.
	BottomRight	Text is vertically aligned at the bottom, and horizontally aligned on the right.
	AlignmentFlag	For internal use only.
	NoWrap	The only flag that can be combines with any of the alignment flags. Specifies that the drawn text should not wrap when the text does not fit in a single line.

See Also

MindFusion.Drawing Namespace

TextureBrush Class

TextureBrush objects represent brushes that use images to fill the interiors of geometric shapes.

Syntax

C# Copy Code

```
public class TextureBrush : Brush
```

Visual Basic Copy Code

```
Public Class TextureBrush
    Inherits Brush
```

Inheritance Hierarchy

System.Object
  MindFusion.Drawing.Brush
    MindFusion.Drawing.TextureBrush

See Also

TextureBrush Members
MindFusion.Drawing Namespace

TextureBrush Members

TextureBrush objects represent brushes that use images to fill the interiors of geometric shapes.

The following tables list the members exposed by the **TextureBrush** type.

Public Constructors

	Name	Description
	TextureBrush	Initializes a new instance of the **TextureBrush** class with the specified image.

Public Properties

	Name	Description
	Image	Gets or sets the **Image** instance associated with this brush.
	Transform	

	WrapMode	Gets or sets a value that indicates the wrap mode for this brush.

Public Methods

	Name	Description
	Clone	Creates an exact copy of this brush.
	Equals	Brush.Equals override.
	GetHashCode	Brush.GetHashCode override.
	ToColor	Brush.ToColor override.

See Also

TextureBrush Class
MindFusion.Drawing Namespace

TextureBrush Constructor

Initializes a new instance of the TextureBrush class with the specified image.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public TextureBrush (
    IImage image
)
```

Visual Basic Copy Code

```
Public New ( _
    image As IImage _
)
```

Parameters

image
The **Image** object with which this brush fills interiors.

See Also

TextureBrush Class
MindFusion.Drawing Namespace

TextureBrush Methods

TextureBrush.Clone Method

Creates an exact copy of this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override Brush Clone ()
```

Visual Basic Copy Code

```
Public Overrides Function Clone () As Brush
```

Return Value

An instance of the TextureBrush class.

See Also

TextureBrush Members
TextureBrush Class
MindFusion.Drawing Namespace

TextureBrush.Equals Method

Brush.Equals override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override bool Equals (
    Brush other
)
```

Visual Basic Copy Code

```
Public Overrides Function Equals( _
    other As Brush _
) As Boolean
```

Parameters

other
The Brush to compare with the current Brush.

Return Value

true if the specified Brush is equal to the current Brush; otherwise, false.

See Also

TextureBrush Members
TextureBrush Class
MindFusion.Drawing Namespace

TextureBrush.GetHashCode Method

Brush.GetHashCode override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override int GetHashCode ()
```

Visual Basic Copy Code

```
Public Overrides Function GetHashCode () As Integer
```

Return Value

See Also

TextureBrush Members
TextureBrush Class
MindFusion.Drawing Namespace

TextureBrush.ToColor Method

Brush.ToColor override.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public override Color ToColor ()
```

Visual Basic Copy Code

```
Public Overrides Function ToColor () As Color
```

Return Value

See Also

TextureBrush Members
TextureBrush Class
MindFusion.Drawing Namespace

TextureBrush Properties

TextureBrush.Image Property

Gets or sets the **Image** instance associated with this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IImage Image { get; set; }
```

Visual Basic Copy Code

```
Public Property Image As IImage
```

Property Value

A **Microsoft.Maui.Graphics.IImage** that represents the image with which this brush fills shapes.

See Also

TextureBrush Members
TextureBrush Class
MindFusion.Drawing Namespace

TextureBrush.Transform Property

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public IMatrix Transform { get; set; }
```

Visual Basic Copy Code

```
Public Property Transform As IMatrix
```

Property Value

See Also

TextureBrush Members
TextureBrush Class
MindFusion.Drawing Namespace

TextureBrush.WrapMode Property

Gets or sets a value that indicates the wrap mode for this brush.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public WrapMode WrapMode { get; set; }
```

Visual Basic Copy Code

```
Public Property WrapMode As WrapMode
```

Property Value

A member of the WrapMode enumeration that specifies how fills drawn with this brush are tiled.

See Also

TextureBrush Members
TextureBrush Class
MindFusion.Drawing Namespace

VerticalAlignment Enumeration

Specifies the vertical alignment of an element.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum VerticalAlignment
```

Visual Basic Copy Code

```
Public Enum VerticalAlignment
```

Members

	Member name	Description
	Top	Indicates that an element is top-aligned to its container.
	Center	Indicates that an element is centered in its container.
	Bottom	Indicates that an element is bottom-aligned to its container.

See Also

MindFusion.Drawing Namespace

WrapMode Enumeration

Specifies how brush content is tiled when it is smaller than the area being filled.

Namespace: MindFusion.Drawing
Assembly: MindFusion.Common

Syntax

C# Copy Code

```
public enum WrapMode
```

Visual Basic Copy Code

```
Public Enum WrapMode
```

Members

	Member name	Description
	Tile	Tiles the brush content.
	TileFlipX	Reverses the brush content horizontally and then tiles it.
	TileFlipY	Reverses the brush content vertically and then tiles it.
	TileFlipXY	Reverses the brush content horizontally and vertically and then tiles it.
	Clamp	The brush content is not tiled.

See Also

MindFusion.Drawing Namespace

MindFusion.Layout Namespace

AnnealLayoutStatistics Class

Contains the accumulated costs for a graph configuration produced by AnnealLayout.

Syntax

C# Copy Code

```
public class AnnealLayoutStatistics
```

Visual Basic	Copy Code
`Public Class AnnealLayoutStatistics`	

Inheritance Hierarchy

System.Object
  MindFusion.Layout.AnnealLayoutStatistics

See Also

AnnealLayoutStatistics Members
MindFusion.Layout Namespace

AnnealLayoutStatistics Members

Contains the accumulated costs for a graph configuration produced by AnnealLayout.

The following tables list the members exposed by the AnnealLayoutStatistics type.

Public Constructors

	Name	Description
	AnnealLayoutStatistics	

Public Properties

	Name	Description
	BoundaryCost	The total distance-to-boundary cost of a graph configuration.
	DistributionCost	The total distribution cost of a graph configuration.
	EdgeCrossingsCost	The total edge crossings cost of a graph configuration.
	EdgeCrossingsFound	The number of edge crossings in the graph configuration.
	EdgeLengthsCost	The total edge lengths cost of a graph configuration.
	NodeEdgeDistance	The total node-to-edge-distance cost of a graph configuration.

Public Methods

	Name	Description
	ToString	Object.ToString override.

See Also

AnnealLayoutStatistics Class
MindFusion.Layout Namespace

AnnealLayoutStatistics Constructor

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C#	Copy Code
`public AnnealLayoutStatistics ()`	

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

AnnealLayoutStatistics Class
MindFusion.Layout Namespace

AnnealLayoutStatistics Methods

AnnealLayoutStatistics.ToString Method

Object.ToString override.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override string ToString ()
```

Visual Basic Copy Code

```
Public Overrides Function ToString () As String
```

Return Value

See Also

AnnealLayoutStatistics Members
AnnealLayoutStatistics Class
MindFusion.Layout Namespace

AnnealLayoutStatistics Properties

AnnealLayoutStatistics.BoundaryCost Property

The total distance-to-boundary cost of a graph configuration.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double BoundaryCost { get; set; }
```

Visual Basic Copy Code

```
Public Property BoundaryCost As Double
```

Property Value

See Also

AnnealLayoutStatistics Members
AnnealLayoutStatistics Class

AnnealLayoutStatistics.DistributionCost Property

The total distribution cost of a graph configuration.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double DistributionCost { get; set; }
```

Visual Basic Copy Code

```
Public Property DistributionCost As Double
```

Property Value

See Also

AnnealLayoutStatistics Members
AnnealLayoutStatistics Class
MindFusion.Layout Namespace

AnnealLayoutStatistics.EdgeCrossingsCost Property

The total edge crossings cost of a graph configuration.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double EdgeCrossingsCost { get; set; }
```

Visual Basic Copy Code

```
Public Property EdgeCrossingsCost As Double
```

Property Value

See Also

AnnealLayoutStatistics Members
AnnealLayoutStatistics Class
MindFusion.Layout Namespace

AnnealLayoutStatistics.EdgeCrossingsFound Property

The number of edge crossings in the graph configuration.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int EdgeCrossingsFound { get; set; }
```

Visual Basic Copy Code

```
Public Property EdgeCrossingsFound As Integer
```

Property Value

See Also

AnnealLayoutStatistics Members
AnnealLayoutStatistics Class
MindFusion.Layout Namespace

AnnealLayoutStatistics.EdgeLengthsCost Property

The total edge lengths cost of a graph configuration.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double EdgeLengthsCost { get; set; }
```

Visual Basic Copy Code

```
Public Property EdgeLengthsCost As Double
```

Property Value

See Also

AnnealLayoutStatistics Members
AnnealLayoutStatistics Class
MindFusion.Layout Namespace

AnnealLayoutStatistics.NodeEdgeDistance Property

The total node-to-edge-distance cost of a graph configuration.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double NodeEdgeDistance { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeEdgeDistance As Double
```

Property Value

See Also

AnnealLayoutStatistics Members
AnnealLayoutStatistics Class
MindFusion.Layout Namespace

AssistantType Enumeration

Indicates the type of an assistant node in a tree.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum AssistantType
```

Visual Basic Copy Code

```
Public Enum AssistantType
```

Members

	Member name	Description
	None	The node is not an assistant.
	Normal	The node is an assistant and its relative position is determined automatically by the layout.
	Left	The node is an assistant and it is always positioned to the left of its parent.
	Right	The node is an assistant and it is always positioned to the right of its parent.

See Also

MindFusion.Layout Namespace

BaseLayout Enumeration

Specifies how OrthogonalRouter should arrange nodes.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum BaseLayout
```

Visual Basic Copy Code

```
Public Enum BaseLayout
```

Members

	Member name	Description
	Layered	Use LayeredLayout to arrange nodes.
	Grid	Use GridLayout to arrange nodes.
	Custom	Use the current node configuration.
	Cascading	Use TreeLayout to arrange nodes.

See Also

ConnectorRules Class

Represents a set of rules that are applied according to the logic identifiers of links.

Syntax

C# Copy Code

```
public class ConnectorRules : LayoutRules
```

Visual Basic Copy Code

```
Public Class ConnectorRules
    Inherits LayoutRules
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.LayoutRules
    MindFusion.Layout.ConnectorRules

See Also

ConnectorRules Members
MindFusion.Layout Namespace

ConnectorRules Members

Represents a set of rules that are applied according to the logic identifiers of links.

The following tables list the members exposed by the ConnectorRules type.

Public Properties

	Name	Description
	DownOriented	Specifies a set of link identifiers for which nodes connected through such links will be placed downwards from already arranged nodes. You can assign a single logic identifier or an array of identifiers to this property.
	HorizontalBendOrientation	Specifies whether links should start with a horizontal segment.
	LeftOriented	Specifies a set of link identifiers for which nodes connected through such links will be placed leftwards from already arranged nodes. You can assign a single logic identifier or an array of identifiers to this property.
	RightOriented	Specifies a set of link identifiers for which nodes connected through such links will be placed rightwards from already arranged nodes. You can assign a single logic identifier or an array of identifiers to this property.
	UpOriented	Specifies a set of link identifiers for which nodes connected via such links will be placed upwards from already arranged nodes. You can assign a single logic identifier or an array of identifiers to this property.

Public Methods

	Name	Description
	AddLayoutRule	Overloaded. (Inherited from LayoutRules.)
	BuildKey	Builds a key - object for the new rule, as a pair of two objects: *obj1* and *obj2*. (Inherited from LayoutRules.)

	GetLayoutRule	If there is rule with key - Key initializes - Value otherwise fails; (Inherited from LayoutRules.)
	ValidateRules	Checks if there is a rule for each key in KeyList (Inherited from LayoutRules.)

See Also

ConnectorRules Class
MindFusion.Layout Namespace

ConnectorRules Properties

ConnectorRules.DownOriented Property

Specifies a set of link identifiers for which nodes connected through such links will be placed downwards from already arranged nodes. You can assign a single logic identifier or an array of identifiers to this property.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Object DownOriented { get; set; }
```

Visual Basic Copy Code

```
Public Property DownOriented As Object
```

Property Value

See Also

ConnectorRules Members
ConnectorRules Class
MindFusion.Layout Namespace

ConnectorRules.HorizontalBendOrientation Property

Specifies whether links should start with a horizontal segment.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool HorizontalBendOrientation { get; set; }
```

Visual Basic Copy Code

```
Public Property HorizontalBendOrientation As Boolean
```

Property Value

See Also

ConnectorRules Members
ConnectorRules Class
MindFusion.Layout Namespace

ConnectorRules.LeftOriented Property

Specifies a set of link identifiers for which nodes connected through such links will be placed leftwards from already arranged nodes. You can assign a single logic identifier or an array of identifiers to this property.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Object LeftOriented { get; set; }
```

Visual Basic Copy Code

```
Public Property LeftOriented As Object
```

Property Value

See Also

ConnectorRules Members
ConnectorRules Class
MindFusion.Layout Namespace

ConnectorRules.RightOriented Property

Specifies a set of link identifiers for which nodes connected through such links will be placed rightwards from already arranged nodes. You can assign a single logic identifier or an array of identifiers to this property.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Object RightOriented { get; set; }
```

Visual Basic Copy Code

```
Public Property RightOriented As Object
```

Property Value

See Also

ConnectorRules Members
ConnectorRules Class
MindFusion.Layout Namespace

ConnectorRules.UpOriented Property

Specifies a set of link identifiers for which nodes connected via such links will be placed upwards from already arranged nodes. You can assign a single logic identifier or an array of identifiers to this property.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Object UpOriented { get; set; }
```

Visual Basic Copy Code

```
Public Property UpOriented As Object
```

Property Value

See Also

ConnectorRules Members
ConnectorRules Class
MindFusion.Layout Namespace

FlowLayoutKeys Class

Defines keys that can be used to define FlowLayout rules.

Syntax

C# Copy Code

```
public class FlowLayoutKeys : SimpleKeys
```

Visual Basic Copy Code

```
Public Class FlowLayoutKeys
    Inherits SimpleKeys
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.SimpleKeys
    MindFusion.Layout.FlowLayoutKeys

See Also

FlowLayoutKeys Members
MindFusion.Layout Namespace

FlowLayoutKeys Members

Defines keys that can be used to define FlowLayout rules.

The following tables list the members exposed by the FlowLayoutKeys type.

Public Constructors

	Name	Description
	FlowLayoutKeys	Initializes a new instance of the **FlowLayoutKeys** class.

Public Fields

	Name	Description
	ConnectorTypes	Specifies a list of connector identifiers. (Inherited from SimpleKeys.)
	FirstGroupLevelNegative	A rule defining which nodes should be placed above the "Control Flow" line. Associate an array of the node types logic identifiers with this key.

		(Inherited from SimpleKeys.)
	FirstGroupLevelPositive	A rule defining which nodes should be placed below the "Control Flow" line. Associate an array of the node types logic identifiers with this key. (Inherited from SimpleKeys.)
	GroupEndNodes	Specifies which nodes should be considered the end of the current sub-group. Associate an array of the node types logic identifiers with this key. (Inherited from SimpleKeys.)
	GroupStartNodes	Specifies which nodes should be considered the start of a new sub-group. Associate an array of the node types logic identifiers with this key. (Inherited from SimpleKeys.)
	MainConnectorOrientation	Specifies the orientation of the "Control Flow" sequence of links, and thus the main orientation for the layout. (Inherited from SimpleKeys.)
	MainSubgroupConnector	Defines which type of connectors can be used as guideline (defining the graph control flow). (Inherited from SimpleKeys.)
	PossibleConnection	Specifies what kind of node connections are allowed. (Inherited from SimpleKeys.)
	PossibleNodeLogic	Specifies a list of identifiers for nodes that should be processed by FlowLayout.
	SemiLevel	Specifies a list of identifiers for nodes that should be placed with some offset from the main sequence line.

Public Properties

	Name	Description
	AllKeys	A list of all keys defined in this class. (Inherited from SimpleKeys.)

Public Methods

	Name	Description
	ValidateRules	Checks if there is a rule for each key in the key list. (Inherited from SimpleKeys.)

See Also

FlowLayoutKeys Class
MindFusion.Layout Namespace

FlowLayoutKeys Fields

FlowLayoutKeys.PossibleNodeLogic Field

Specifies a list of identifiers for nodes that should be processed by FlowLayout.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string PossibleNodeLogic
```

Visual Basic Copy Code

```
Public Shared PossibleNodeLogic As String
```

See Also

FlowLayoutKeys Members
FlowLayoutKeys Class
MindFusion.Layout Namespace

FlowLayoutKeys.SemiLevel Field

Specifies a list of identifiers for nodes that should be placed with some offset from the main sequence line.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string SemiLevel
```

Visual Basic Copy Code

```
Public Shared SemiLevel As String
```

See Also

FlowLayoutKeys Members
FlowLayoutKeys Class
MindFusion.Layout Namespace

FlowLayoutKeys Constructor

Initializes a new instance of the FlowLayoutKeys class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public FlowLayoutKeys ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

FlowLayoutKeys Class
MindFusion.Layout Namespace

FlowLayoutRules Class

A class representing a set of rules that define the behavior of the **FlowLayout** class.

Syntax

C# Copy Code

```
public class FlowLayoutRules : LayoutRules
```

Visual Basic	Copy Code

```
Public Class FlowLayoutRules
    Inherits LayoutRules
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.LayoutRules
    MindFusion.Layout.FlowLayoutRules

See Also

FlowLayoutRules Members
MindFusion.Layout Namespace

FlowLayoutRules Members

A class representing a set of rules that define the behavior of the **FlowLayout** class.

The following tables list the members exposed by the FlowLayoutRules type.

Public Constructors

	Name	Description
	FlowLayoutRules	Initializes a new instance of the **FlowLayoutRules** class.

Public Properties

	Name	Description
	ConnectorRules	A set of rules that are applied according to the logic identifiers of links.
	NodeRules	A set of rules that are applied according to the logic identifiers of node pairs.
	PreferredRules	Specifies which set of rules to use.

Public Methods

	Name	Description
	AddBendRule	Overloaded.
	AddIncomingBendRule	Overloaded.
	AddLayoutRule	Overloaded. (Inherited from LayoutRules.)
	AddOutgoingBendRule	Overloaded.
	BuildKey	Builds a key - object for the new rule, as a pair of two objects: *obj1* and *obj2*. (Inherited from LayoutRules.)
	GetLayoutRule	If there is rule with key - Key initializes - Value otherwise fails; (Inherited from LayoutRules.)
	UseRuleManagedLinkBend	Specifies whether FlowLayout should use custom link-bending rules specified through the various Add*BendRule methods.
	ValidateRules	Checks if there is a rule for each key in KeyList (Inherited from LayoutRules.)

See Also

FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules Constructor

Initializes a new instance of the FlowLayoutRules class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public FlowLayoutRules ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules Methods

FlowLayoutRules.AddBendRule Method

Overload List

Name	Description
FlowLayoutRules.AddBendRule (String, String, OrderDirection)	Adds a rule for the orientation of the starting or ending segment of a link according to the type of the node connected to that link.
FlowLayoutRules.AddBendRule (String, String, OrderDirection[])	Adds a rule for the orientation of the starting or ending segments of a link according to the type of the node connected to that link.
FlowLayoutRules.AddBendRule (String, String[], OrderDirection)	Adds a rule for the orientation of the starting or ending segment of a link according to the type of the node connected to that link.
FlowLayoutRules.AddBendRule (String, String[], OrderDirection[])	Adds a rule for the orientation of the starting or ending segments of a link according to the type of the node connected to that link.
FlowLayoutRules.AddBendRule (String[], String, OrderDirection)	Adds a rule for the orientation of the starting or ending segment of a link according to the type of the node connected to that link.
FlowLayoutRules.AddBendRule (String[], String, OrderDirection[])	Adds a rule for the orientation of the starting or ending segments of a link according to the type of the node connected to that link.

See Also

FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddBendRule Method (String, String, OrderDirection)

Adds a rule for the orientation of the starting or ending segment of a link according to the type of the node connected to that link.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddBendRule (
    string nodeType,
    string linkType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddBendRule( _
    nodeType As String, _
    linkType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

nodeType
The logic identifier of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

direction
An OrderDirection value defining the orientation of the links starting or ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddBendRule Method (String, String, OrderDirection[])

Adds a rule for the orientation of the starting or ending segments of a link according to the type of the node connected to that link.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddBendRule (
    string nodeType,
    string linkType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddBendRule( _
    nodeType As String, _
    linkType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

nodeType
The logic identifier of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting or ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddBendRule Method (String, String[], OrderDirection)

Adds a rule for the orientation of the starting or ending segment of a link according to the type of the node connected to that link.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddBendRule (
    string nodeType,
    string[] linkTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddBendRule( _
    nodeType As String, _
    linkTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

nodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

direction
An OrderDirection value defining the orientation of the links starting or ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddBendRule Method (String, String[], OrderDirection[])

Adds a rule for the orientation of the starting or ending segments of a link according to the type of the node connected to that link.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddBendRule (
    string nodeType,
    string[] linkTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddBendRule( _
    nodeType As String, _
    linkTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

nodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting or ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddBendRule Method (String[], String, OrderDirection)

Adds a rule for the orientation of the starting or ending segment of a link according to the type of the node connected to that link.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddBendRule (
    string[] nodeTypes,
    string linkType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddBendRule( _
    nodeTypes() As String, _
    linkType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

nodeTypes
The logic identifiers of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

direction
An OrderDirection value defining the orientation of the links starting or ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddBendRule Method (String[], String, OrderDirection[])

Adds a rule for the orientation of the starting or ending segments of a link according to the type of the node connected to that link.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddBendRule (
    string[] nodeTypes,
    string linkType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddBendRule( _
    nodeTypes() As String, _
    linkType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

nodeTypes
The logic identifiers of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting or ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method

Overload List

Name	Description
FlowLayoutRules.AddIncomingBendRule (String, String, OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination node.
FlowLayoutRules.AddIncomingBendRule (String, String, OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination node.
FlowLayoutRules.AddIncomingBendRule (String, String, String, OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String, String, String, OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String, String, String[], OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String, String, String[], OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String, String[], OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination node.
FlowLayoutRules.AddIncomingBendRule (String, String[], OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination node.
FlowLayoutRules.AddIncomingBendRule (String, String[], String, OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String, String[], String, OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String, String[], String[], OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String, String[], String[], OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String[], String, OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination node.
FlowLayoutRules.AddIncomingBendRule (String[], String, OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination node.
FlowLayoutRules.AddIncomingBendRule (String[], String, String, OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String[], String, String, OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String[], String, String[], OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String[], String, String[], OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String[], String[], OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination node.
FlowLayoutRules.AddIncomingBendRule (String[], String[], OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination node.
FlowLayoutRules.AddIncomingBendRule (String[], String[], String, OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String[], String[], String, OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.
FlowLayoutRules.AddIncomingBendRule (String[], String[], String[], OrderDirection)	Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.

FlowLayoutRules.AddIncomingBendRule (String[], String[], String[], OrderDirection[])	Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.

See Also

FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String, OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string linkType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String, OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string linkType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String, String, OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string linkType,
    string originNodeType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkType As String, _
    originNodeType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

originNodeType
The logic identifier of nodes this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String, String, OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string linkType,
    string originNodeType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkType As String, _
    originNodeType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

originNodeType
The logic identifier of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String, String[], OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string linkType,
    string[] originNodeTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkType As String, _
    originNodeTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String, String[], OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string linkType,
```

```
    string[] originNodeTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkType As String, _
    originNodeTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String[], OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string[] linkTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

direction

An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String[], OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string[] linkTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String[], String, OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string[] linkTypes,
    string originNodeType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkTypes() As String, _
    originNodeType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

originNodeType
The logic identifier of nodes this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String[], String, OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string[] linkTypes,
    string originNodeType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkTypes() As String, _
    originNodeType As String, _
```

```
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

originNodeType
The logic identifier of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String[], String[], OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string[] linkTypes,
    string[] originNodeTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkTypes() As String, _
    originNodeTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String, String[], String[], OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string destNodeType,
    string[] linkTypes,
    string[] originNodeTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeType As String, _
    linkTypes() As String, _
    originNodeTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String, OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string linkType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String, OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string linkType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String, String, OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string linkType,
    string originNodeType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkType As String, _
    originNodeType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

originNodeType
The logic identifier of nodes this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String, String, OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string linkType,
    string originNodeType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkType As String, _
    originNodeType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

originNodeType
The logic identifier of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String, String[], OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string linkType,
```

```
    string[] originNodeTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkType As String, _
    originNodeTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String, String[], OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string linkType,
    string[] originNodeTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkType As String, _
    originNodeTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String[], OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string[] linkTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String[], OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string[] linkTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String[], String, OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string[] linkTypes,
    string originNodeType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkTypes() As String, _
    originNodeType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

originNodeType
The logic identifier of nodes this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String[], String, OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string[] linkTypes,
    string originNodeType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkTypes() As String, _
    originNodeType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

originNodeType
The logic identifier of nodes this rule should be applied to.

directions

An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String[], String[], OrderDirection)

Adds a rule for the orientation of the ending segment of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string[] linkTypes,
    string[] originNodeTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkTypes() As String, _
    originNodeTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's ending segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddIncomingBendRule Method (String[], String[], String[], OrderDirection[])

Adds a rule for the orientation of the ending segments of an incoming link according to the type of the link's destination and origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddIncomingBendRule (
    string[] destNodeTypes,
    string[] linkTypes,
    string[] originNodeTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddIncomingBendRule( _
    destNodeTypes() As String, _
    linkTypes() As String, _
    originNodeTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links ending segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddIncomingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method

Overload List

Name	Description
FlowLayoutRules.AddOutgoingBendRule (String, String, OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String, String, OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String, String, String, OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

FlowLayoutRules.AddOutgoingBendRule (String, String, String, OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String, String, String[], OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String, String, String[], OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String, String[], OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String, String[], OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String, String[], String, OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String, String[], String, OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String, String[], String[], OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String, String[], String[], OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String, OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String, OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String, String, OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String, String, OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String, String[], OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String, String[], OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String[], OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String[], OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String[], String, OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String[], String, OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String[], String[], OrderDirection)	Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.
FlowLayoutRules.AddOutgoingBendRule (String[], String[], String[], OrderDirection[])	Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

See Also

FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String, OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string linkType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
    linkType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule to be applied to.

linkType
The logic identifier of links this rule will be applied to.

direction
An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String, OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string linkType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
```

```
    linkType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String, String, OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string linkType,
    string destNodeType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
    linkType As String, _
    destNodeType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule will be applied to.

linkType
The logic identifier of links this rule will be applied to.

destNodeType
The logic identifier of nodes this rule will be applied to.

direction
An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String, String, OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string linkType,
    string destNodeType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
    linkType As String, _
    destNodeType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

destNodeType
The logic identifier of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String, String[], OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string linkType,
    string[] destNodeTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
    linkType As String, _
    destNodeTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule will be applied to.

linkType
The logic identifier of links this rule will be applied to.

destNodeTypes
The logic identifiers of nodes this rule will be applied to.

direction
An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String, String[], OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string linkType,
    string[] destNodeTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
    linkType As String, _
    destNodeTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule should be applied to.

linkType
The logic identifier of links this rule should be applied to.

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String[], OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string[] linkTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
    linkTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String[], OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string[] linkTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
    linkTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String[], String, OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string[] linkTypes,
    string destNodeType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
    linkTypes() As String, _
    destNodeType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

destNodeType
The logic identifier of nodes this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String[], String, OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string[] linkTypes,
    string destNodeType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
    linkTypes() As String, _
    destNodeType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

destNodeType
The logic identifier of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String[], String[], OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string[] linkTypes,
    string[] destNodeTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
    linkTypes() As String, _
    destNodeTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule to apply to.

linkTypes
The logic identifiers of links this rule to be applied to.

destNodeTypes
The logic identifiers of nodes this rule to be applied to.

direction
An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String, String[], String[], OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string originNodeType,
    string[] linkTypes,
    string[] destNodeTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeType As String, _
    linkTypes() As String, _
    destNodeTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeType
The logic identifier of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String, OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string linkType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
    linkType As String, _
```

```
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes

The logic identifiers of nodes this rule to be applied to.

linkType

The logic identifier of links this rule to be applied to.

direction

An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String, OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string linkType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
    linkType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes

The logic identifiers of nodes this rule to be applied to.

linkType

The logic identifier of links this rule should be applied to.

directions

An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String, String, OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string linkType,
    string destNodeType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
    linkType As String, _
    destNodeType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes
The logic identifiers of nodes this rule to apply.

linkType
The logic identifier of links this rule to apply.

destNodeType
The logic identifier of nodes this rule will be applied to.

direction
An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String, String, OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string linkType,
    string destNodeType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
    linkType As String, _
    destNodeType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes
The logic identifiers of nodes this rule to be applied to.

linkType
The logic identifier of links this rule to be applied to.

destNodeType
The logic identifier of nodes this rule to be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String, String[], OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string linkType,
    string[] destNodeTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
    linkType As String, _
    destNodeTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes

The logic identifiers of nodes this rule to be applied to.

linkType
The logic identifier of links this rule to be applied to.

destNodeTypes
The logic identifiers of nodes this rule to be applied to.

direction
An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String, String[], OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string linkType,
    string[] destNodeTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
    linkType As String, _
    destNodeTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes
The logic identifiers of nodes this rule to be applied to.

linkType
The logic identifier of links this rule to be applied to.

destNodeTypes
The logic identifiers of nodes this rule to be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String[], OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string[] linkTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
    linkTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes
The logic identifiers of nodes this rule to be applied to.

linkTypes
The logic identifiers of links this rule to be applied to.

direction
An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String[], OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string[] linkTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
```

```
    linkTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkTypes
The logic identifier of links this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String[], String, OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string[] linkTypes,
    string destNodeType,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
    linkTypes() As String, _
    destNodeType As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

destNodeType
The logic identifier of nodes this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String[], String, OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string[] linkTypes,
    string destNodeType,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
    linkTypes() As String, _
    destNodeType As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

destNodeType
The logic identifier of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String[], String[], OrderDirection)

Adds a rule for the orientation of the starting segment of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string[] linkTypes,
    string[] destNodeTypes,
    OrderDirection direction
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
    linkTypes() As String, _
    destNodeTypes() As String, _
    direction As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

direction
An OrderDirection value defining the orientation of the link's starting segment.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.AddOutgoingBendRule Method (String[], String[], String[], OrderDirection[])

Adds a rule for the orientation of the starting segments of an outgoing link according to the type of the link's origin node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddOutgoingBendRule (
    string[] originNodeTypes,
    string[] linkTypes,
    string[] destNodeTypes,
    OrderDirection[] directions
)
```

Visual Basic Copy Code

```
Public Function AddOutgoingBendRule( _
    originNodeTypes() As String, _
    linkTypes() As String, _
    destNodeTypes() As String, _
    directions() As OrderDirection _
) As Boolean
```

Parameters

originNodeTypes
The logic identifiers of nodes this rule should be applied to.

linkTypes
The logic identifiers of links this rule should be applied to.

destNodeTypes
The logic identifiers of nodes this rule should be applied to.

directions
An array of OrderDirection values defining the orientation of the links starting segments.

Return Value

true if this rule has been added successfully; otherwise, **false**.

See Also

AddOutgoingBendRule Method Overload List
FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.UseRuleManagedLinkBend Method

Specifies whether FlowLayout should use custom link-bending rules specified through the various Add*BendRule methods.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool UseRuleManagedLinkBend (
    bool ruleManaged
)
```

Visual Basic Copy Code

```
Public Function UseRuleManagedLinkBend( _
    ruleManaged As Boolean _
) As Boolean
```

Parameters

ruleManaged
true to use custom link-bending rules; otherwise, **false**.

Return Value

true if this rule has been set successfully; otherwise, **false**.

See Also

FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules Properties

FlowLayoutRules.ConnectorRules Property

A set of rules that are applied according to the logic identifiers of links.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public ConnectorRules ConnectorRules { get; set; }
```

Visual Basic Copy Code

```
Public Property ConnectorRules As ConnectorRules
```

Property Value

See Also

FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.NodeRules Property

A set of rules that are applied according to the logic identifiers of node pairs.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public NodeRules NodeRules { get; set; }
```

Visual Basic Copy Code

```
Public Property NodeRules As NodeRules
```

Property Value

See Also

FlowLayoutRules Members
FlowLayoutRules Class
MindFusion.Layout Namespace

FlowLayoutRules.PreferredRules Property

Specifies which set of rules to use.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public PreferredRules PreferredRules { get; set; }
```

Visual Basic Copy Code

```
Public Property PreferredRules As PreferredRules
```

Property Value

See Also

FlowLayoutTraits Enumeration

Values used as keys in FlowLayout rules.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum FlowLayoutTraits
```

Visual Basic Copy Code

```
Public Enum FlowLayoutTraits
```

Members

	Member name	Description
	PossibleConnection	For internal use.
	OrderingRule	Specified the preferred order for nodes.
	NodeDirection	For internal use.
	NodeDistance	Specifies the distance left between this node and the next in the sequence.
	LaneNode	Specifies that FlowLayout should treat a node as a lane.
	LogicID	Specifies a string used to identify items in FlowLayout rules.
	Lane	Specifies the lane to which a node should be assigned.
	VerticalSubgroupsAlignment	A boolean value specifying the alignment of nodes within a group.
	StandardLinkBend	For internal use.
	InLink	Identifies possible incoming links in FlowLayout rules.
	OutLink	Identifies possible outgoing links in FlowLayout rules.
	NotSpecified	For internal use.
	VerticalDisconnectedGroupsAlignment	A boolean value specifying the alignment of nodes within a group.

See Also

MindFusion.Layout Namespace

HierarchicalLayoutTraits Enumeration

Indicates item traits specific to hierarchical layouts. Members of this enumeration can be used as keys in the DiagramItem.LayoutTraits hashtable.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum HierarchicalLayoutTraits
```

Visual Basic Copy Code

```
Public Enum HierarchicalLayoutTraits
```

Members

	Member name	Description
	Level	The number associated with this key specifies the index of the level of the item in the level-planar graph.

See Also

MindFusion.Layout Namespace

LayeredLayoutTraits Enumeration

Indicates item traits specific to layered layouts. Members of this enumeration can be used as keys in the DiagramItem.LayoutTraits hashtable.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum LayeredLayoutTraits
```

Visual Basic Copy Code

```
Public Enum LayeredLayoutTraits
```

Members

	Member name	Description
	Layer	The number associated with this key specifies the index of the layer the layout should attempt to place the node to. Specifying a negative number indicates a layer starting from the last. For example, -1 would indicate the last layer, -2 - the layer before the last, and so on.

See Also

MindFusion.Layout Namespace

LayoutRules Class

Summary description for **LayoutRules**.

Syntax

C# Copy Code

```
public class LayoutRules
```

Visual Basic Copy Code

```
Public Class LayoutRules
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.LayoutRules
    MindFusion.Layout.ConnectorRules
    MindFusion.Layout.FlowLayoutRules
    MindFusion.Layout.NodeRules
    MindFusion.Layout.SimpleRules

See Also

LayoutRules Members
MindFusion.Layout Namespace

LayoutRules Members

Summary description for LayoutRules.

The following tables list the members exposed by the **LayoutRules** type.

Public Constructors

	Name	Description
	LayoutRules	Initializes a new instance of the LayoutRules class.

Public Methods

	Name	Description
	AddLayoutRule	Overloaded.
	BuildKey	Builds a key - object for the new rule, as a pair of two objects: *obj1* and *obj2*.
	GetLayoutRule	If there is rule with key - Key initializes - Value otherwise fails;
	ValidateRules	Checks if there is a rule for each key in KeyList

See Also

LayoutRules Class
MindFusion.Layout Namespace

LayoutRules Constructor

Initializes a new instance of the LayoutRules class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public LayoutRules ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

LayoutRules Class
MindFusion.Layout Namespace

LayoutRules Methods

LayoutRules.AddLayoutRule Method

Overload List

Name	Description
LayoutRules.AddLayoutRule (Object, Object)	Adds or sets a new rule, depending on if the rule with key - Key already exists or no;
LayoutRules.AddLayoutRule (Object, Object, Object)	Builds a key as a pair of the objects Object1 and Object2 and adds or sets a new rule, with this key and value - Value;

See Also

LayoutRules Members
LayoutRules Class
MindFusion.Layout Namespace

LayoutRules.AddLayoutRule Method (Object, Object)

Adds or sets a new rule, depending on if the rule with key - Key already exists or no;

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddLayoutRule (
    Object Key,
    Object Value
)
```

Visual Basic Copy Code

```
Public Function AddLayoutRule( _
    Key As Object, _
    Value As Object _
) As Boolean
```

Parameters

Key

Value

Return Value

See Also

AddLayoutRule Method Overload List
LayoutRules Members
LayoutRules Class
MindFusion.Layout Namespace

LayoutRules.AddLayoutRule Method (Object, Object, Object)

Builds a key as a pair of the objects Object1 and Object2 and adds or sets a new rule, with this key and value - Value;

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool AddLayoutRule (
    Object Object1,
    Object Object2,
    Object Value
)
```

Visual Basic Copy Code

```
Public Function AddLayoutRule( _
    Object1 As Object, _
    Object2 As Object, _
    Value As Object _
) As Boolean
```

Parameters

Object1

Object2

Value

Return Value

See Also

AddLayoutRule Method Overload List
LayoutRules Members
LayoutRules Class
MindFusion.Layout Namespace

LayoutRules.BuildKey Method

Builds a key - object for the new rule, as a pair of two objects: *obj1* and *obj2*.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Object BuildKey (
    Object obj1,
    Object obj2
)
```

Visual Basic Copy Code

```
Public Function BuildKey( _
    obj1 As Object, _
    obj2 As Object _
) As Object
```

Parameters

obj1

obj2

Return Value

See Also

LayoutRules.GetLayoutRule Method

If there is rule with key - Key initializes - Value otherwise fails;

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public virtual bool GetLayoutRule (
    Object Key,
    out Object Value
)
```

Visual Basic Copy Code

```
Public Overridable Function GetLayoutRule( _
    Key As Object, _
    <System.Runtime.InteropServices.Out()> ByRef Value As Object _
) As Boolean
```

Parameters

Key

Value

Return Value

See Also

LayoutRules Members
LayoutRules Class
MindFusion.Layout Namespace

LayoutRules.ValidateRules Method

Checks if there is a rule for each key in KeyList

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool ValidateRules (
    ArrayList KeyList
)
```

Visual Basic Copy Code

```
Public Function ValidateRules( _
    KeyList As ArrayList _
) As Boolean
```

Parameters

KeyList

Return Value

See Also

LayoutRules Members
LayoutRules Class
MindFusion.Layout Namespace

NodeRules Class

Represents a set of rules that are applied according to the logic identifiers of node pairs.

Syntax

C# Copy Code

```
public class NodeRules : LayoutRules
```

Visual Basic Copy Code

```
Public Class NodeRules
    Inherits LayoutRules
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.LayoutRules
    MindFusion.Layout.NodeRules

See Also

NodeRules Members
MindFusion.Layout Namespace

NodeRules Members

Represents a set of rules that are applied according to the logic identifiers of node pairs.

The following tables list the members exposed by the NodeRules type.

Public Properties

	Name	Description
	FirstGroupLevelNegative	A rule defining which nodes should be placed above the "Control Flow" line. Assign an array of the node types logic identifiers to this property.
	FirstGroupLevelPositive	A rule defining which nodes should be placed below the "Control Flow" line. Assign an array of the node types logic identifiers to this property.
	GroupEndNodes	Specifies which nodes should be considered the end of the current sub-group. Assign an array of the node types logic identifiers to this property.
	GroupStartNodes	Specifies which nodes should be considered the start of a new sub-group. Assign an array of the node types logic identifiers to this property.
	MainConnectorId	Specifies which type of connectors define the control flow in the process represented by the arranged diagram.
	MainConnectorOrientation	Specifies the orientation of the "Control Flow" sequence of links, and thus the main orientation for the layout.
	VerticalDisconnectedGroupsAlignment	Specifies whether disconnected subgroups from the same lane should be aligned and arranged vertically.
	VerticalSubgroupsAlignment	Specifies whether connected subgroups from the same lane should be aligned and arranged vertically.

Public Methods

	Name	Description
	AddLayoutRule	Overloaded. (Inherited from LayoutRules.)
	BuildKey	Builds a key - object for the new rule, as a pair of two objects: *obj1* and *obj2*. (Inherited from LayoutRules.)
	GetLayoutRule	If there is rule with key - Key initializes - Value otherwise fails; (Inherited from LayoutRules.)
	ValidateRules	Checks if there is a rule for each key in KeyList (Inherited from LayoutRules.)

See Also

NodeRules Class
MindFusion.Layout Namespace

NodeRules Properties

NodeRules.FirstGroupLevelNegative Property

A rule defining which nodes should be placed above the "Control Flow" line. Assign an array of the node types logic identifiers to this property.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Object FirstGroupLevelNegative { get; set; }
```

Visual Basic Copy Code

```
Public Property FirstGroupLevelNegative As Object
```

Property Value

See Also

NodeRules Members
NodeRules Class
MindFusion.Layout Namespace

NodeRules.FirstGroupLevelPositive Property

A rule defining which nodes should be placed below the "Control Flow" line. Assign an array of the node types logic identifiers to this property.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Object FirstGroupLevelPositive { get; set; }
```

Visual Basic Copy Code

```
Public Property FirstGroupLevelPositive As Object
```

Property Value

See Also

NodeRules Members
NodeRules Class
MindFusion.Layout Namespace

NodeRules.GroupEndNodes Property

Specifies which nodes should be considered the end of the current sub-group. Assign an array of the node types logic identifiers to this property.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Object GroupEndNodes { get; set; }
```

Visual Basic Copy Code

```
Public Property GroupEndNodes As Object
```

Property Value

See Also

NodeRules Members
NodeRules Class
MindFusion.Layout Namespace

NodeRules.GroupStartNodes Property

Specifies which nodes should be considered the start of a new sub-group. Assign an array of the node types logic identifiers to this property.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Object GroupStartNodes { get; set; }
```

Visual Basic Copy Code

```
Public Property GroupStartNodes As Object
```

Property Value

See Also

NodeRules Members
NodeRules Class
MindFusion.Layout Namespace

NodeRules.MainConnectorId Property

Specifies which type of connectors define the control flow in the process represented by the arranged diagram.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public string MainConnectorId { get; set; }
```

Visual Basic Copy Code

```
Public Property MainConnectorId As String
```

Property Value

See Also

NodeRules Members
NodeRules Class
MindFusion.Layout Namespace

NodeRules.MainConnectorOrientation Property

Specifies the orientation of the "Control Flow" sequence of links, and thus the main orientation for the layout.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SimpleOrientation MainConnectorOrientation { get; set; }
```

Visual Basic Copy Code

```
Public Property MainConnectorOrientation As SimpleOrientation
```

Property Value

See Also

NodeRules Members
NodeRules Class
MindFusion.Layout Namespace

NodeRules.VerticalDisconnectedGroupsAlignment Property

Specifies whether disconnected subgroups from the same lane should be aligned and arranged vertically.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool VerticalDisconnectedGroupsAlignment { get; set; }
```

Visual Basic Copy Code

```
Public Property VerticalDisconnectedGroupsAlignment As Boolean
```

Property Value

See Also

NodeRules Members
NodeRules Class
MindFusion.Layout Namespace

NodeRules.VerticalSubgroupsAlignment Property

Specifies whether connected subgroups from the same lane should be aligned and arranged vertically.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool VerticalSubgroupsAlignment { get; set; }
```

Visual Basic Copy Code

```
Public Property VerticalSubgroupsAlignment As Boolean
```

Property Value

See Also

NodeRules Members
NodeRules Class
MindFusion.Layout Namespace

OrderDirection Enumeration

Values controlling the direction of link segments in FlowLayout rules.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum OrderDirection
```

Visual Basic Copy Code

```
Public Enum OrderDirection
```

Members

	Member name	Description
	Leftward	Specifies leftward link direction.
	Rightward	Specifies rightward link direction.
	Upward	Specifies upward link direction.
	Downward	Specifies downward link direction.

	NotSpecified	Unspecified link direction.

See Also

MindFusion.Layout Namespace

Pair Class

Represents a coupling between two arbitrary objects.

Syntax

C# Copy Code

```
public class Pair : IComparable
```

Visual Basic Copy Code

```
Public Class Pair
    Implements IComparable
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.Pair

See Also

Pair Members
MindFusion.Layout Namespace

Pair Members

Represents a coupling between two arbitrary objects.

The following tables list the members exposed by the Pair type.

Public Constructors

	Name	Description
	Pair	Overloaded.

Public Properties

	Name	Description
	FirstObject	Gets or sets the first pair object.
	PairElements	Gets the elements of the pair as an arraylist.
	SecondObject	Gets or sets the second pair object.

Public Methods

	Name	Description
	CompareTo	Implements IComparable.CompareTo.
	Equals	Object.Equals override.
	GetHashCode	Object.GetHashCode override.

See Also

Pair Class
MindFusion.Layout Namespace

Pair Constructor

Overload List

Name	Description
Pair ()	Initializes a new instance of the Pair class.
Pair (Object, Object)	Initializes a new instance of the Pair class.

See Also

Pair Class
MindFusion.Layout Namespace

Pair Constructor ()

Initializes a new instance of the Pair class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Pair ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

Pair Constructor Overload List
Pair Class
MindFusion.Layout Namespace

Pair Constructor (Object, Object)

Initializes a new instance of the Pair class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Pair (
    Object obj1,
    Object obj2
)
```

Visual Basic Copy Code

```
Public New ( _
    obj1 As Object, _
    obj2 As Object _
)
```

Parameters

obj1

obj2

See Also

Pair Constructor Overload List
Pair Class
MindFusion.Layout Namespace

Pair Methods

Pair.CompareTo Method

Implements IComparable.CompareTo.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int CompareTo (
    Object obj
)
```

Visual Basic Copy Code

```
Public Function CompareTo( _
    obj As Object _
) As Integer
```

Parameters

obj

Return Value

See Also

Pair Members
Pair Class
MindFusion.Layout Namespace

Pair.Equals Method

Object.Equals override.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override bool Equals (
    Object obj
)
```

Visual Basic Copy Code

```
Public Overrides Function Equals( _
    obj As Object _
) As Boolean
```

Parameters

obj

Return Value

See Also

Pair Members
Pair Class
MindFusion.Layout Namespace

Pair.GetHashCode Method

Object.GetHashCode override.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public override int GetHashCode ()
```

Visual Basic Copy Code

```
Public Overrides Function GetHashCode () As Integer
```

Return Value

See Also

Pair Members
Pair Class
MindFusion.Layout Namespace

Pair Properties

Pair.FirstObject Property

Gets or sets the first pair object.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Object FirstObject { get; set; }
```

Visual Basic Copy Code

```
Public Property FirstObject As Object
```

Property Value

See Also

Pair Members
Pair Class
MindFusion.Layout Namespace

Pair.PairElements Property

Gets the elements of the pair as an arraylist.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public ArrayList PairElements { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property PairElements As ArrayList
```

Property Value

See Also

Pair Members
Pair Class
MindFusion.Layout Namespace

Pair.SecondObject Property

Gets or sets the second pair object.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Object SecondObject { get; set; }
```

Visual Basic Copy Code

```
Public Property SecondObject As Object
```

Property Value

See Also

Pair Members
Pair Class
MindFusion.Layout Namespace

PartitionMethod Enumeration

Identifies graph partitioning methods supported by CompositeLayout.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum PartitionMethod
```

Visual Basic Copy Code

```
Public Enum PartitionMethod
```

Members

	Member name	Description
	BiconnectedComponents	Partition graphs by their biconnected components.
	ShortestPath	Partition graphs by node paths distance.
	Custom	Custom partitioning.
	LinkDensity	

See Also

MindFusion.Layout Namespace

PathNode Class

Represents a path node, used by the link routing algorithm.

Syntax

C# Copy Code

```
public class PathNode : IComparable
```

Visual Basic Copy Code

```
Public Class PathNode
    Implements IComparable
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.PathNode

See Also

PathNode Members
MindFusion.Layout Namespace

PathNode Members

Represents a path node, used by the link routing algorithm.

The following tables list the members exposed by the PathNode type.

Public Constructors

	Name	Description
	PathNode	Initializes a new instance of the PathNode class.

Public Methods

	Name	Description
	CompareTo	IComparable.CompareTo implementation.

See Also

PathNode Class
MindFusion.Layout Namespace

PathNode Constructor

Initializes a new instance of the PathNode class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public PathNode (
    int x,
    int y
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Integer, _
    y As Integer _
)
```

Parameters

x

y

See Also

PathNode Class
MindFusion.Layout Namespace

PathNode Methods

PathNode.CompareTo Method

IComparable.CompareTo implementation.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int CompareTo (
    Object obj
)
```

Visual Basic Copy Code

```
Public Function CompareTo( _
    obj As Object _
) As Integer
```

Parameters

obj

Return Value

See Also

PathNode Members
PathNode Class
MindFusion.Layout Namespace

Point3D Structure

Represents the coordinates of a point in a 3-dimensional space.

Syntax

C# Copy Code

```
public struct Point3D
```

Visual Basic Copy Code

```
Public Structure Point3D
```

See Also

Point3D Members
MindFusion.Layout Namespace

Point3D Members

Represents the coordinates of a point in a 3-dimensional space.

The following tables list the members exposed by the Point3D type.

Public Constructors

	Name	Description
	Point3D	Initializes a new instance of the Point3D class.

Public Properties

	Name	Description
	X	Gets or sets the X coordinate of this point.
	Y	Gets or sets the Y coordinate of this point.
	Z	Gets or sets the Z coordinate of this point.

See Also

Point3D Structure
MindFusion.Layout Namespace

Point3D Constructor

Initializes a new instance of the Point3D class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Point3D (
    double x,
    double y,
    double z
)
```

Visual Basic Copy Code

```
Public New ( _
    x As Double, _
    y As Double, _
    z As Double _
)
```

Parameters

x

y

z

See Also

Point3D Structure
MindFusion.Layout Namespace

Point3D Properties

Point3D.X Property

Gets or sets the X coordinate of this point.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double X { get; set; }
```

Visual Basic Copy Code

```
Public Property X As Double
```

Property Value

See Also

Point3D Members
Point3D Structure
MindFusion.Layout Namespace

Point3D.Y Property

Gets or sets the Y coordinate of this point.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double Y { get; set; }
```

Visual Basic Copy Code

```
Public Property Y As Double
```

Property Value

See Also

Point3D Members
Point3D Structure
MindFusion.Layout Namespace

Point3D.Z Property

Gets or sets the Z coordinate of this point.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public double Z { get; set; }
```

Visual Basic Copy Code

```
Public Property Z As Double
```

Property Value

See Also

Point3D Members
Point3D Structure
MindFusion.Layout Namespace

PointList Class

Syntax

C# Copy Code

```
public class PointList : List<Point>
```

Visual Basic Copy Code

```
Public Class PointList
    Inherits List(Of Point)
```

Inheritance Hierarchy

System.Object
  System.Collections.Generic.List(Point)
    MindFusion.Layout.PointList

See Also

PointList Members
MindFusion.Layout Namespace

PointList Members

The following tables list the members exposed by the PointList type.

Public Constructors

	Name	Description
	PointList	Overloaded.

Public Properties

	Name	Description
	First	Gets or sets the first point in the collection.
	Last	Gets or sets the last point in the collection.

Public Methods

	Name	Description
	AddRange	
	Clone	Creates a shallow copy of this collection.

See Also

PointList Class
MindFusion.Layout Namespace

PointList Constructor

Overload List

Name	Description
PointList ()	Initializes a new instance of the PointList class.
PointList (ICollection)	Initializes a new instance of the PointList class with the specified point values.
PointList (Int32)	Initializes a new instance of the PointList class.

See Also

PointList Class
MindFusion.Layout Namespace

PointList Constructor ()

Initializes a new instance of the PointList class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public PointList ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

PointList Constructor Overload List
PointList Class
MindFusion.Layout Namespace

PointList Constructor (ICollection)

Initializes a new instance of the PointList class with the specified point values.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public PointList (
    ICollection points
)
```

Visual Basic Copy Code

```
Public New ( _
    points As ICollection _
)
```

Parameters

points

See Also

PointList Constructor Overload List
PointList Class
MindFusion.Layout Namespace

PointList Constructor (Int32)

Initializes a new instance of the PointList class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public PointList (
    int size
)
```

Visual Basic Copy Code

```
Public New ( _
    size As Integer _
)
```

Parameters

size

See Also

PointList Constructor Overload List
PointList Class
MindFusion.Layout Namespace

PointList Methods

PointList.AddRange Method

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public void AddRange (
    ICollection points
)
```

Visual Basic Copy Code

```
Public Sub AddRange( _
    points As ICollection _
)
```

Parameters

points

See Also

PointList Members
PointList Class
MindFusion.Layout Namespace

PointList.Clone Method

Creates a shallow copy of this collection.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public PointList Clone ()
```

Visual Basic Copy Code

```
Public Function Clone () As PointList
```

Return Value

See Also

PointList Members
PointList Class
MindFusion.Layout Namespace

PointList Properties

PointList.First Property

Gets or sets the first point in the collection.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Point First { get; set; }
```

Visual Basic Copy Code

```
Public Property First As Point
```

Property Value

Remarks

An IndexOutOfRangeException will be thrown if the collection is empty.

See Also

PointList Members
PointList Class
MindFusion.Layout Namespace

PointList.Last Property

Gets or sets the last point in the collection.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Point Last { get; set; }
```

Visual Basic

Copy Code

```
Public Property Last As Point
```

Property Value

Remarks

An IndexOutOfRangeException will be thrown if the collection is empty.

See Also

PointList Members
PointList Class
MindFusion.Layout Namespace

PreferredRules Enumeration

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum PreferredRules
```

Visual Basic Copy Code

```
Public Enum PreferredRules
```

Members

	Member name	Description
	ConnectorRules	Use connector rules.
	NodeRules	Use node rules.

See Also

MindFusion.Layout Namespace

Routable Interface

Syntax

C# Copy Code

```
public interface Routable
```

Visual Basic Copy Code

```
Public Interface Routable
```

Inheritors

MindFusion.Diagramming.DiagramLink

See Also

Routable Members

The following tables list the members exposed by the Routable type.

Public Methods

	Name	Description
	ApplyRoute	
	CreateRouteDescriptor	

See Also

Routable Interface
MindFusion.Layout Namespace

Routable Methods

Routable.ApplyRoute Method

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
void ApplyRoute (
    List<PointI> routePoints,
    RouteDescriptor routeDescriptor,
    PointList resultPoints
)
```

Visual Basic Copy Code

```
Sub ApplyRoute( _
    routePoints As List(Of PointI), _
    routeDescriptor As RouteDescriptor, _
    resultPoints As PointList _
)
```

Parameters

routePoints

routeDescriptor

resultPoints

See Also

Routable Members
Routable Interface
MindFusion.Layout Namespace

Routable.CreateRouteDescriptor Method

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
RouteDescriptor CreateRouteDescriptor ()
```

Visual Basic Copy Code

```
Function CreateRouteDescriptor () As RouteDescriptor
```

Return Value

See Also

Routable Members
Routable Interface
MindFusion.Layout Namespace

RouteDescriptor Class

Syntax

C# Copy Code

```
public class RouteDescriptor
```

Visual Basic Copy Code

```
Public Class RouteDescriptor
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.RouteDescriptor

See Also

RouteDescriptor Members
MindFusion.Layout Namespace

RouteDescriptor Members

The following tables list the members exposed by the RouteDescriptor type.

Public Constructors

	Name	Description
	RouteDescriptor	

Public Fields

	Name	Description
	dNearest	
	end	
	oNearest	

	routeBounds	
	routeHeuristics	
	routingGrid	
	routingOptions	
	start	

See Also

RouteDescriptor Class
MindFusion.Layout Namespace

RouteDescriptor Fields

RouteDescriptor.dNearest Field

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int dNearest
```

Visual Basic Copy Code

```
Public dNearest As Integer
```

See Also

RouteDescriptor Members
RouteDescriptor Class
MindFusion.Layout Namespace

RouteDescriptor.end Field

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public PointI end
```

Visual Basic Copy Code

```
Public end As PointI
```

See Also

RouteDescriptor Members
RouteDescriptor Class
MindFusion.Layout Namespace

RouteDescriptor.oNearest Field

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public int oNearest
```

Visual Basic Copy Code

```
Public oNearest As Integer
```

See Also

RouteDescriptor Members
RouteDescriptor Class
MindFusion.Layout Namespace

RouteDescriptor.routeBounds Field

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public Rect routeBounds
```

Visual Basic Copy Code

```
Public routeBounds As Rect
```

See Also

RouteDescriptor Members
RouteDescriptor Class
MindFusion.Layout Namespace

RouteDescriptor.routeHeuristics Field

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public RouteHeuristics routeHeuristics
```

Visual Basic Copy Code

```
Public routeHeuristics As RouteHeuristics
```

See Also

RouteDescriptor Members
RouteDescriptor Class

RouteDescriptor.routingGrid Field

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public RoutingGrid routingGrid
```

Visual Basic Copy Code

```
Public routingGrid As RoutingGrid
```

See Also

RouteDescriptor Members
RouteDescriptor Class
MindFusion.Layout Namespace

RouteDescriptor.routingOptions Field

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public RoutingOptions routingOptions
```

Visual Basic Copy Code

```
Public routingOptions As RoutingOptions
```

See Also

RouteDescriptor Members
RouteDescriptor Class
MindFusion.Layout Namespace

RouteDescriptor.start Field

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public PointI start
```

Visual Basic Copy Code

```
Public start As PointI
```

See Also

RouteDescriptor Members
RouteDescriptor Class
MindFusion.Layout Namespace

RouteDescriptor Constructor

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public RouteDescriptor ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

RouteDescriptor Class
MindFusion.Layout Namespace

RouteHeuristics Delegate

The link routing algorithm uses a heuristic function of this type in its prediction steps.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public delegate int RouteHeuristics (
    Point start,
    Point end
)
```

Visual Basic Copy Code

```
Public Delegate Function RouteHeuristics( _
    start As Point, _
    end As Point _
) As Integer
```

Parameters

start
A Point specifying the first of the points, between which the distance is calculated.

end
A Point specifying the first of the points, between which the distance is calculated.

Return Value

An integer value indicating the calculated distance between the specified points.

Remarks

Functions of this type can be assigned to the RouteHeuristics property of the RoutingOptions class.

See Also

MindFusion.Layout Namespace

RouteHeuristics Delegate

The link routing algorithm uses a heuristic function of this type in its prediction steps.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public delegate int RouteHeuristics (
    PointI start,
    PointI end
)
```

Visual Basic Copy Code

```
Public Delegate Function RouteHeuristics( _
    start As PointI, _
    end As PointI _
) As Integer
```

Parameters

start
A Point specifying the first of the points, between which the distance is calculated.

end
A Point specifying the first of the points, between which the distance is calculated.

Return Value

An integer value indicating the calculated distance between the specified points.

Remarks

Functions of this type can be assigned to the RouteHeuristics property of the RoutingOptions class.

See Also

MindFusion.Layout Namespace

RoutingGrid Interface

Syntax

C# Copy Code

```
public interface RoutingGrid
```

Visual Basic Copy Code

```
Public Interface RoutingGrid
```

See Also

RoutingGrid Members
MindFusion.Layout Namespace

RoutingGrid Members

The following tables list the members exposed by the RoutingGrid type.

Public Properties

	Name	Description
	ColumnCount	
	RowCount	

Public Methods

	Name	Description
	GetClosedGrid	Returns the grid of closed path nodes.
	GetCostGrid	Returns the grid of costs assigned to diagram locations.
	GetOpenGrid	Returns the grid of open path nodes.

See Also

RoutingGrid Interface
MindFusion.Layout Namespace

RoutingGrid Methods

RoutingGrid.GetClosedGrid Method

Returns the grid of closed path nodes.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
PathNode[,] GetClosedGrid ()
```

Visual Basic Copy Code

```
Function GetClosedGrid () As PathNode[,]
```

Return Value

See Also

RoutingGrid Members
RoutingGrid Interface
MindFusion.Layout Namespace

RoutingGrid.GetCostGrid Method

Returns the grid of costs assigned to diagram locations.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
Byte[,] GetCostGrid ()
```

Visual Basic Copy Code

```
Function GetCostGrid () As Byte[,]
```

Return Value

See Also

RoutingGrid Members
RoutingGrid Interface
MindFusion.Layout Namespace

RoutingGrid.GetOpenGrid Method

Returns the grid of open path nodes.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
PathNode[,] GetOpenGrid ()
```

Visual Basic Copy Code

```
Function GetOpenGrid () As PathNode[,]
```

Return Value

See Also

RoutingGrid Members
RoutingGrid Interface
MindFusion.Layout Namespace

RoutingGrid Properties

RoutingGrid.ColumnCount Property

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
int ColumnCount { get; }
```

Visual Basic Copy Code

```
ReadOnly Property ColumnCount As Integer
```

Property Value

See Also

RoutingGrid Members
RoutingGrid Interface
MindFusion.Layout Namespace

RoutingGrid.RowCount Property

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
int RowCount { get; }
```

Visual Basic Copy Code

```
ReadOnly Property RowCount As Integer
```

Property Value

See Also

RoutingGrid Members
RoutingGrid Interface
MindFusion.Layout Namespace

RoutingOptions Interface

Allows customization of various aspects of the link routing algorithm.

Syntax

C# Copy Code

```
public interface RoutingOptions
```

Visual Basic Copy Code

```
Public Interface RoutingOptions
```

Inheritors

MindFusion.Diagramming.RoutingOptions

See Also

RoutingOptions Members
MindFusion.Layout Namespace

RoutingOptions Members

Allows customization of various aspects of the link routing algorithm.

The following tables list the members exposed by the RoutingOptions type.

Public Properties

	Name	Description
	GridSize	Gets or sets the size of the pieces of the routing grid.

	IgnoreAnchoring	For internal use.
	LengthCost	Gets or sets a value added to the total cost of a route for each piece of the routing grid occupied by the route.
	NodeVicinityCost	Gets or sets the size of a buffer zone around a node for which routes get a penalty cost if they pass through it.
	TurnCost	Gets or sets a value added to the total cost of a route if it makes a turn.

See Also

RoutingOptions Interface
MindFusion.Layout Namespace

RoutingOptions Properties

RoutingOptions.GridSize Property

Gets or sets the size of the pieces of the routing grid.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
double GridSize { get; set; }
```

Visual Basic Copy Code

```
Property GridSize As Double
```

Property Value

See Also

RoutingOptions Members
RoutingOptions Interface
MindFusion.Layout Namespace

RoutingOptions.IgnoreAnchoring Property

For internal use.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
bool IgnoreAnchoring { get; }
```

Visual Basic Copy Code

```
ReadOnly Property IgnoreAnchoring As Boolean
```

Property Value

See Also

RoutingOptions Members
RoutingOptions Interface
MindFusion.Layout Namespace

RoutingOptions.LengthCost Property

Gets or sets a value added to the total cost of a route for each piece of the routing grid occupied by the route.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
byte LengthCost { get; }
```

Visual Basic Copy Code

```
ReadOnly Property LengthCost As Byte
```

Property Value

See Also

RoutingOptions Members
RoutingOptions Interface
MindFusion.Layout Namespace

RoutingOptions.NodeVicinityCost Property

Gets or sets the size of a buffer zone around a node for which routes get a penalty cost if they pass through it.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
byte NodeVicinityCost { get; }
```

Visual Basic Copy Code

```
ReadOnly Property NodeVicinityCost As Byte
```

Property Value

See Also

RoutingOptions Members
RoutingOptions Interface
MindFusion.Layout Namespace

RoutingOptions.TurnCost Property

Gets or sets a value added to the total cost of a route if it makes a turn.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
byte TurnCost { get; }
```

Visual Basic Copy Code

```
ReadOnly Property TurnCost As Byte
```

Property Value

See Also

RoutingOptions Members
RoutingOptions Interface
MindFusion.Layout Namespace

SimpleKeys Class

Defines keys that can be used to define FlowLayout rules.

Syntax

C# Copy Code

```
public class SimpleKeys
```

Visual Basic Copy Code

```
Public Class SimpleKeys
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.SimpleKeys
    MindFusion.Layout.FlowLayoutKeys

See Also

SimpleKeys Members
MindFusion.Layout Namespace

SimpleKeys Members

Defines keys that can be used to define FlowLayout rules.

The following tables list the members exposed by the SimpleKeys type.

Public Constructors

	Name	Description
	SimpleKeys	Initializes a new instance of the SimpleKeys class.

Public Fields

	Name	Description
	ConnectorTypes	Specifies a list of connector identifiers.
	FirstGroupLevelNegative	A rule defining which nodes should be placed above the "Control Flow" line. Associate an array of the node types logic identifiers with this key.
	FirstGroupLevelPositive	A rule defining which nodes should be placed below the "Control Flow" line. Associate an array of the node types logic identifiers with this key.

	GroupEndNodes	Specifies which nodes should be considered the end of the current sub-group. Associate an array of the node types logic identifiers with this key.
	GroupStartNodes	Specifies which nodes should be considered the start of a new sub-group. Associate an array of the node types logic identifiers with this key.
	MainConnectorOrientation	Specifies the orientation of the "Control Flow" sequence of links, and thus the main orientation for the layout.
	MainSubgroupConnector	Defines which type of connectors can be used as guideline (defining the graph control flow).
	PossibleConnection	Specifies what kind of node connections are allowed.

Public Properties

	Name	Description
	AllKeys	A list of all keys defined in this class.

Public Methods

	Name	Description
	ValidateRules	Checks if there is a rule for each key in the key list.

See Also

SimpleKeys Class
MindFusion.Layout Namespace

SimpleKeys Fields

SimpleKeys.ConnectorTypes Field

Specifies a list of connector identifiers.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string ConnectorTypes
```

Visual Basic Copy Code

```
Public Shared ConnectorTypes As String
```

See Also

SimpleKeys Members
SimpleKeys Class
MindFusion.Layout Namespace

SimpleKeys.FirstGroupLevelNegative Field

A rule defining which nodes should be placed above the "Control Flow" line. Associate an array of the node types logic identifiers with this key.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string FirstGroupLevelNegative
```

Visual Basic Copy Code

```
Public Shared FirstGroupLevelNegative As String
```

See Also

SimpleKeys Members
SimpleKeys Class
MindFusion.Layout Namespace

SimpleKeys.FirstGroupLevelPositive Field

A rule defining which nodes should be placed below the "Control Flow" line. Associate an array of the node types logic identifiers with this key.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string FirstGroupLevelPositive
```

Visual Basic Copy Code

```
Public Shared FirstGroupLevelPositive As String
```

See Also

SimpleKeys Members
SimpleKeys Class
MindFusion.Layout Namespace

SimpleKeys.GroupEndNodes Field

Specifies which nodes should be considered the end of the current sub-group. Associate an array of the node types logic identifiers with this key.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string GroupEndNodes
```

Visual Basic Copy Code

```
Public Shared GroupEndNodes As String
```

See Also

SimpleKeys Members
SimpleKeys Class
MindFusion.Layout Namespace

SimpleKeys.GroupStartNodes Field

Specifies which nodes should be considered the start of a new sub-group. Associate an array of the node types logic identifiers with this key.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string GroupStartNodes
```

Visual Basic Copy Code

```
Public Shared GroupStartNodes As String
```

See Also

SimpleKeys Members
SimpleKeys Class
MindFusion.Layout Namespace

SimpleKeys.MainConnectorOrientation Field

Specifies the orientation of the "Control Flow" sequence of links, and thus the main orientation for the layout.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string MainConnectorOrientation
```

Visual Basic Copy Code

```
Public Shared MainConnectorOrientation As String
```

See Also

SimpleKeys Members
SimpleKeys Class
MindFusion.Layout Namespace

SimpleKeys.MainSubgroupConnector Field

Defines which type of connectors can be used as guideline (defining the graph control flow).

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string MainSubgroupConnector
```

Visual Basic Copy Code

```
Public Shared MainSubgroupConnector As String
```

See Also

SimpleKeys Members
SimpleKeys Class
MindFusion.Layout Namespace

SimpleKeys.PossibleConnection Field

Specifies what kind of node connections are allowed.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string PossibleConnection
```

Visual Basic Copy Code

```
Public Shared PossibleConnection As String
```

See Also

SimpleKeys Members
SimpleKeys Class
MindFusion.Layout Namespace

SimpleKeys Constructor

Initializes a new instance of the SimpleKeys class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SimpleKeys ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SimpleKeys Class
MindFusion.Layout Namespace

SimpleKeys Methods

SimpleKeys.ValidateRules Method

Checks if there is a rule for each key in the key list.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public bool ValidateRules (
    LayoutRules lr
)
```

Visual Basic Copy Code

```
Public Function ValidateRules( _
    lr As LayoutRules _
) As Boolean
```

Parameters

lr

Return Value

See Also

SimpleKeys Members
SimpleKeys Class
MindFusion.Layout Namespace

SimpleKeys Properties

SimpleKeys.AllKeys Property

A list of all keys defined in this class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public ArrayList AllKeys { get; }
```

Visual Basic Copy Code

```
Public ReadOnly Property AllKeys As ArrayList
```

Property Value

See Also

SimpleKeys Members
SimpleKeys Class
MindFusion.Layout Namespace

SimpleLinkLogic Class

Contains link identifiers for the predefined SimpleRules set of rules.

Syntax

C# Copy Code

```
public class SimpleLinkLogic
```

Visual Basic Copy Code

```
Public Class SimpleLinkLogic
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.SimpleLinkLogic

See Also

SimpleLinkLogic Members
MindFusion.Layout Namespace

SimpleLinkLogic Members

Contains link identifiers for the predefined SimpleRules set of rules.

The following tables list the members exposed by the SimpleLinkLogic type.

Public Constructors

	Name	Description
	SimpleLinkLogic	Initializes a new instance of the SimpleLinkLogic class.

Public Fields

	Name	Description
	Master	Identifier for the main connector type.
	NotSpecified	Unspecified connector type.
	Slave	Identifier for the secondary connector type.

See Also

SimpleLinkLogic Class
MindFusion.Layout Namespace

SimpleLinkLogic Fields

SimpleLinkLogic.Master Field

Identifier for the main connector type.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string Master
```

Visual Basic Copy Code

```
Public Shared Master As String
```

See Also

SimpleLinkLogic Members
SimpleLinkLogic Class
MindFusion.Layout Namespace

SimpleLinkLogic.NotSpecified Field

Unspecified connector type.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string NotSpecified
```

Visual Basic Copy Code

```
Public Shared NotSpecified As String
```

See Also

SimpleLinkLogic Members
SimpleLinkLogic Class
MindFusion.Layout Namespace

SimpleLinkLogic.Slave Field

Identifier for the secondary connector type.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string Slave
```

Visual Basic Copy Code

```
Public Shared Slave As String
```

See Also

SimpleLinkLogic Members
SimpleLinkLogic Class
MindFusion.Layout Namespace

SimpleLinkLogic Constructor

Initializes a new instance of the SimpleLinkLogic class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SimpleLinkLogic ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SimpleLinkLogic Class
MindFusion.Layout Namespace

SimpleNodeLogic Class

Contains node identifiers for the predefined SimpleRules set of rules.

Syntax

C# Copy Code

```
public class SimpleNodeLogic
```

Visual Basic Copy Code

```
Public Class SimpleNodeLogic
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.SimpleNodeLogic

See Also

SimpleNodeLogic Members
MindFusion.Layout Namespace

SimpleNodeLogic Members

Contains node identifiers for the predefined SimpleRules set of rules.

The following tables list the members exposed by the SimpleNodeLogic type.

Public Constructors

	Name	Description
	SimpleNodeLogic	

Public Fields

	Name	Description
	Action	Identifier for action nodes.
	End	Identifier for group end nodes.
	NotSpecified	Unspecified node type.
	Start	Identifier for group start nodes.

See Also

SimpleNodeLogic Class
MindFusion.Layout Namespace

SimpleNodeLogic Fields

SimpleNodeLogic.Action Field

Identifier for action nodes.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string Action
```

Visual Basic Copy Code

```
Public Shared Action As String
```

See Also

SimpleNodeLogic Members
SimpleNodeLogic Class
MindFusion.Layout Namespace

SimpleNodeLogic.End Field

Identifier for group end nodes.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string End
```

Visual Basic Copy Code

```
Public Shared End As String
```

See Also

SimpleNodeLogic Members
SimpleNodeLogic Class
MindFusion.Layout Namespace

SimpleNodeLogic.NotSpecified Field

Unspecified node type.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string NotSpecified
```

Visual Basic Copy Code

```
Public Shared NotSpecified As String
```

See Also

SimpleNodeLogic Members
SimpleNodeLogic Class
MindFusion.Layout Namespace

SimpleNodeLogic.Start Field

Identifier for group start nodes.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public static string Start
```

Visual Basic Copy Code

```
Public Shared Start As String
```

See Also

SimpleNodeLogic Members
SimpleNodeLogic Class
MindFusion.Layout Namespace

SimpleNodeLogic Constructor

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SimpleNodeLogic ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SimpleNodeLogic Class
MindFusion.Layout Namespace

SimpleOrientation Enumeration

Specifies the preferred orientation for the main connector type in FlowLayout.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum SimpleOrientation
```

Visual Basic — Copy Code

```
Public Enum SimpleOrientation
```

Members

	Member name	Description
	LeftToRight	Specifies left-to-right orientation.
	RightToLeft	Specifies right-to-left orientation.
	BottomToTop	Specifies bottom-to-top orientation.
	TopToBottom	Specifies top-to-bottom orientation.
	None	Unspecified orientation.

See Also

MindFusion.Layout Namespace

SimpleRules Class

Class of simple rules applicable to the Flowlayout algorithm.

Syntax

C# — Copy Code

```
public class SimpleRules : LayoutRules
```

Visual Basic — Copy Code

```
Public Class SimpleRules
    Inherits LayoutRules
```

Inheritance Hierarchy

System.Object
  MindFusion.Layout.LayoutRules
    MindFusion.Layout.SimpleRules

See Also

SimpleRules Members
MindFusion.Layout Namespace

SimpleRules Members

Class of simple rules applicable to the Flowlayout algorithm.

The following tables list the members exposed by the SimpleRules type.

Public Constructors

	Name	Description
	SimpleRules	Overloaded.

Public Properties

	Name	Description
	EndSubgpouNode	Defines which type of nodes will be used as a "sub group end" node;
	MainConnectorOrientation	A key defining the space orientation of a "Control Flow" and respectively for the layout;
	MainSubgpouConnector	Defines which type of connectors can be used as guideline (defining the graph control flow).
	StartSubgpouNode	Defines which type of nodes will be used as a "sub group" start node.

Public Methods

	Name	Description
	AddLayoutRule	Overloaded. (Inherited from LayoutRules.)
	BuildKey	Builds a key - object for the new rule, as a pair of two objects: *obj1* and *obj2*. (Inherited from LayoutRules.)
	GetLayoutRule	If there is rule with key - Key initializes - Value otherwise fails; (Inherited from LayoutRules.)
	ValidateRules	Checks if there is a rule for each key in KeyList (Inherited from LayoutRules.)

See Also

SimpleRules Class
MindFusion.Layout Namespace

SimpleRules Constructor

Overload List

Name	Description
SimpleRules ()	Initializes a new instance of the SimpleRules class.
SimpleRules (ArrayList, ArrayList, String)	Initializes a new instance of the SimpleRules class.
SimpleRules (ArrayList, ArrayList, String, SimpleOrientation)	Initializes a new instance of the SimpleRules class.

See Also

SimpleRules Class
MindFusion.Layout Namespace

SimpleRules Constructor ()

Initializes a new instance of the SimpleRules class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SimpleRules ()
```

Visual Basic Copy Code

```
Public Sub New ()
```

See Also

SimpleRules Constructor Overload List
SimpleRules Class
MindFusion.Layout Namespace

SimpleRules Constructor (ArrayList, ArrayList, String)

Initializes a new instance of the SimpleRules class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SimpleRules (
    ArrayList strtSubGroupNodeType_ls,
    ArrayList endSubGroupNodeType_ls,
    string mainConnector
)
```

Visual Basic Copy Code

```
Public New ( _
    strtSubGroupNodeType_ls As ArrayList, _
    endSubGroupNodeType_ls As ArrayList, _
    mainConnector As String _
)
```

Parameters

strtSubGroupNodeType_ls

endSubGroupNodeType_ls

mainConnector

See Also

SimpleRules Constructor Overload List
SimpleRules Class
MindFusion.Layout Namespace

SimpleRules Constructor (ArrayList, ArrayList, String, SimpleOrientation)

Initializes a new instance of the SimpleRules class.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SimpleRules (
    ArrayList strtSubGroupNodeType_ls,
    ArrayList endSubGroupNodeType_ls,
    string mainConnector,
    SimpleOrientation mainConnectorOrdrDrctn
)
```

Visual Basic

Copy Code

```
Public New ( _
    strtSubGroupNodeType_ls As ArrayList, _
    endSubGroupNodeType_ls As ArrayList, _
    mainConnector As String, _
    mainConnectorOrdrDrctn As SimpleOrientation _
)
```

Parameters

strtSubGroupNodeType_ls

endSubGroupNodeType_ls

mainConnector

mainConnectorOrdrDrctn

See Also

SimpleRules Constructor Overload List
SimpleRules Class
MindFusion.Layout Namespace

SimpleRules Properties

SimpleRules.EndSubgproupNode Property

Defines which type of nodes will be used as a "sub group end" node;

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public ArrayList EndSubgproupNode { get; set; }
```

Visual Basic Copy Code

```
Public Property EndSubgproupNode As ArrayList
```

Property Value

See Also

SimpleRules Members
SimpleRules Class
MindFusion.Layout Namespace

SimpleRules.MainConnectorOrientation Property

A key defining the space orientation of a "Control Flow" and respectively for the layout;

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public SimpleOrientation MainConnectorOrientation { get; set; }
```

Visual Basic Copy Code

```
Public Property MainConnectorOrientation As SimpleOrientation
```

Property Value

See Also

SimpleRules Members
SimpleRules Class
MindFusion.Layout Namespace

SimpleRules.MainSubgproupConnector Property

Defines which type of connectors can be used as guideline (defining the graph control flow).

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public string MainSubgproupConnector { get; set; }
```

Visual Basic Copy Code

```
Public Property MainSubgproupConnector As String
```

Property Value

See Also

SimpleRules Members
SimpleRules Class
MindFusion.Layout Namespace

SimpleRules.StartSubgproupNode Property

Defines which type of nodes will be used as a "sub group" start node.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public ArrayList StartSubgproupNode { get; set; }
```

Visual Basic Copy Code

```
Public Property StartSubgproupNode As ArrayList
```

Property Value

See Also

SimpleRules Members
SimpleRules Class
MindFusion.Layout Namespace

SpringLayoutMethod Enumeration

Specifies what energy model SpringLayout should use for the simulation.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum SpringLayoutMethod
```

Visual Basic Copy Code

```
Public Enum SpringLayoutMethod
```

Members

	Member name	Description
	Classic	Use Hooke's law of elasticity to treat links as springs, and Coulomb's inverse square law to implement repulsion between nodes.
	FruchtermanReingold	Use artificial attraction forces for links and repulsion forces for nodes that cancel each other at about the distance specified in NodeDistance, along with a simulated cooling schedule used for fine-tuning. Based on the method of Thomas Fruchterman and Edward Reingold.

See Also

MindFusion.Layout Namespace

SpringLayoutTraits Enumeration

Indicates item traits specific to SpringLayout. Members of this enumeration can be used as keys in the DiagramItem.LayoutTraits hashtable.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum SpringLayoutTraits
```

Visual Basic Copy Code

```
Public Enum SpringLayoutTraits
```

Members

	Member name	Description
	Frozen	If the LayoutTraits value associated with a node through this key is true, SpringLayout considers the repulsion forces exerted by the node to other nodes, but ignores forces acting on the node and keeps it in place.

See Also

MindFusion.Layout Namespace

SwimlaneLayoutTraits Enumeration

Indicates item traits specific to swimlane layouts. Members of this enumeration can be used as keys in the DiagramItem.LayoutTraits hashtable.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum SwimlaneLayoutTraits
```

Visual Basic Copy Code

```
Public Enum SwimlaneLayoutTraits
```

Members

	Member name	Description
	Lane	Specifies the lane to which a node should be assigned.
	SortOrder	Specifies custom sort order of nodes in the same lane.

See Also

MindFusion.Layout Namespace

TreeLayoutTraits Enumeration

Indicates item traits specific to tree layouts. Members of this enumeration can be used as keys in the DiagramItem.LayoutTraits hashtable.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum TreeLayoutTraits
```

Visual Basic Copy Code

```
Public Enum TreeLayoutTraits
```

Members

	Member name	Description
	Assistant	If the LayoutTraits value associated with a node through this key is **true**, centered TreeLayout treats this node as an "assistant" during the arrangement. If the value is **false** or is not present, the node is treated as "subordinate". The associated value can also be of type AssistantType if it is required to specify the position of the assistant node relative to its parent. This value is considered when the EnableAssistants property of **TreeLayout** is set to **true**.

See Also

MindFusion.Layout Namespace

TreeMapLayoutTraits Enumeration

Indicates item traits specific to tree map layouts. Members of this enumeration can be used as keys in the DiagramItem.LayoutTraits hashtable.

Namespace: MindFusion.Layout
Assembly: MindFusion.Diagramming.Layout

Syntax

C# Copy Code

```
public enum TreeMapLayoutTraits
```

Visual Basic Copy Code

```
Public Enum TreeMapLayoutTraits
```

Members

	Member name	Description
	HeaderSize	The header size of a master node in a tree-map layout.

See Also

MindFusion.Layout Namespace

Table of Predefined Shapes

Shape	Identifier	Common Uses
	Actor	A user An actor in use case model
	Alternative	An alternative in SDL An extraction in flowcharts Storage in quality management
	Arrow1	Use as desired
	Arrow2	Use as desired
	Arrow3	Transportation in quality management A reference point
	Arrow4	Use as desired
	Arrow5	Use as desired
	Arrow6	Use as desired

	Arrow7	A signal Electronic transmission of data
	Arrow8	Use as desired
	Arrow9	Use as desired
	BeginLoop	Begin of a loop
	BpmnComplex	BPMN Gateways: Complex Decision/Merge
	BpmnDataBasedXor	BPMN Gateways: Data-Based Exclusive Decision/Merge (XOR)
	BpmnEndCancel	BPMN Events: Cancel (End)
	BpmnEndCompensation	BPMN Events: Compensation (End)
	BpmnEndError	BPMN Events: Error (End)
	BpmnEndLink	BPMN Events: Link (End)
	BpmnEndMessage	BPMN Events: Message (End)
	BpmnEndMultiple	BPMN Events: Multiple (End)
	BpmnEndTerminate	BPMN Events: Terminate (End)
	BpmnEventBasedXor	BPMN Gateways: Event-Based Exclusive Decision/Merge (XOR)
	BpmnInclusive	BPMN Gateways: Inclusive Decision/Merge (OR)

	BpmnIntermediateCancel	BPMN Events: Cancel (Intermediate)
	BpmnIntermediateCompensation	BPMN Events: Compensation (Intermediate)
	BpmnIntermediateError	BPMN Events: Error (Intermediate)
	BpmnIntermediateLink	BPMN Events: Link (Intermediate)
	BpmnIntermediateMessage	BPMN Events: Message (Intermediate)
	BpmnIntermediateMultiple	BPMN Events: Multiple (Intermediate)
	BpmnIntermediateRule	BPMN Events: Rule (Intermediate)
	BpmnIntermediateTimer	BPMN Events: Timer (Intermediate)
	BpmnParallelFork	BPMN Gateways: Parallel Fork/Join (AND)
	BpmnStartLink	BPMN Events: Link (Start)
	BpmnStartMessage	BPMN Events: Message (Start)
	BpmnStartMultiple	BPMN Events: Multiple (Start)
	BpmnStartRule	BPMN Events: Rule (Start)
	BpmnStartTimer	BPMN Events: Timer (Start)
	Cloud	Object instance in data flow diagrams

	Collate	A collation in flowcharts
	ConeDown	Use as desired
	ConeUp	Use as desired
	ConnectedIssues	Connected issues in quality management
	CreateRequest	Request to create in SDL
	Cross	Use as desired
	Cube	Use as desired
	Cylinder	Use as desired
	Database	A database in flowcharts
	DataTransmition	Electronic transmission of data
	DDelay	Use as desired
	Decagon	Use as desired
	Decision	Decision in flowcharts A comparison in flowcharts Relationship in data flow diagrams
	Decision2	A decision in SDL
	Delay	A delay in process diagrams
	DInDelay	Use as desired
	DirectAccessStorage	A direct access storage

	DiskStorage	Data stored on disk in flowcharts
	Display	Data that is displayed for human use
	DividedEvent	An event
	DividedProcess	A process or document in flowcharts Entity in data flow diagrams
	Document	A document
	Donut	Use as desired
	DOutDelay	Use as desired
	Ellipse	A cross-reference within the same page of a flowchart Process in data flow diagrams State in data flow diagrams A continuation of the flowchart in SDL An use case Operation in quality management
	EndLoop	End of a loop
	ExternalOrganization	An external organization
	ExternalProcess	An external process
	File	A file
	FramedRectangle	Use as desired
	Gate	A gateway
	Heart	Use as desired
	Heptagon	Use as desired

	Hourglass	Use as desired
	Input	Data that is input by manual means
	InternalStorage	An internal storage
	Interrupt	An interrupt
	Interrupt2	An interrupt
	IsoProcess	Iso9000 process
	Junction	A summing junction A return in SDL
	Lightning	Use as desired
	LinedDocument	A document in flowcharts
	ManualOperation	Operation performed by a human
	Merge	A merger in flowcharts Inbound goods in quality management Non-electronic file Stop state in data flow diagrams
	MessageFromUser	A message from user side in SDL
	MessageToUser	A message to user side in SDL
	Microform	A microform
	MicroformProcessing	A microform processing
	MicroformRecording	A microform recording
	MultiDocument	A multiple document in flowcharts

	MultiProc	A multiple process or document in flowcharts
	Octagon	Use as desired
	OfflineStorage	An offline storage
	OffpageConnection	A cross-reference to another page of a flowchart
	OffpageReference	A cross-reference to another page of a flowchart A foreign module
	Or	An 'or' statement
	Pentagon	Use as desired
	Plaque	Use as desired
	PrimitiveFromCall	A primitive from call control in SDL
	PrimitiveToCall	A primitive to call control in SDL
	Procedure	Procedure in SDL
	Process	A process or document in flowcharts A file or data storage
	Prysm	Use as desired
	PunchedCard	A punched card
	PyramidDown	Use as desired
	PyramidUp	Use as desired
	Rectangle	A metric operation 'External interactor' in data flow diagrams Inspection in quality management

	RightTriangle	Use as desired
	RoundRect	A process or item in flowcharts An event An issue in quality management
	RSave	Input or output data in flowcharts
	Save	A save in SDL Input or output data in flowcharts
	Sort	A sort operation
	Sphere	Use as desired
	Star	Use as desired
	Star4Pointed	Use as desired
	Star5Pointed	Use as desired
	Star6Pointed	Use as desired
	Star7Pointed	Use as desired
	Star16Pointed	Use as desired
	Star24Pointed	Use as desired
	Star32Pointed	Use as desired
	Start	Start of a program flow A starting point in SDL
	Tape	Sequentially accessible data
	Terminator	Start or end of a program flow

	TransmittalTape	A transmittal tape
	Trapezoid	Use as desired

Online Resources

Video Tutorials

- Collaborative drawing of diagrams with SignalR and the ASP.NET MVC flowchart control

Blog Posts

- Collaborative drawing with MindFusion.Diagramming and SignalR

Technical Support

Detailed information about everything connected with software products offered by MindFusion - online documentation, demos, information about forthcoming products can be found at https://mindfusion.dev/. The site is updated every day, so make sure you visit it regularly to catch up with the newest from us.

Apart from that, MindFusion employs a dedicated and technically proficient support team. This guarantees that all your questions and inquiries will be answered in quick and professional manner. They can be reached by e-mail at mailto:support@mindfusion.eu or by phone at +359 889 199 729.

You can post a question about MvcDiagram on the discussion board:

https://mindfusion.dev/Forum/YaBB.pl?board=mvcdg_disc

or visit our help desk site:

https://mindfusion.dev/HelpDesk/index.php

We will be happy to hear any suggestions and recommendations you might have regarding MvcDiagram. We do take in account the opinion of every single person who has shown interest in our products.

MindFusion Products

MindFusion offers various programming components for different platforms. Below is a list of our products with links to their sections on our website as well link to download a trial version. Clients that have already purchased a product use an increasing discount for each subsequent one.

Diagramming	Links
Diagramming for **WPF**	Product Information Download
Diagramming for **WinForms**	Product Information Download
Diagramming for **Java Swing**	Product Information Download
Diagramming for **JavaScript**	Product Information Download
Diagramming for **ASP.NET**	Product Information Download
Diagramming for **ASP.NET MVC**	Product Information Download
Diagramming for **Xamarin**	Product Information Download

Diagramming for **UWP**	Product Information Download
Diagramming for **Android**	Product Information Download
DiagramViewer for **iOS/Cocoa Touch**	Product Information Download
Diagramming for **ActiveX**	Product Information Download
Diagramming for **Silverlight**	Product Information Download

Charting	Links
Charting for **WPF**	Product Information Download
Charting for **WinForms**	Product Information Download
Charting for **Java Swing**	Product Information Download
Charting for **JavaScript**	Product Information Download
Charting for **ASP.NET**	Product Information Download
Charting for **Xamarin**	Product Information Download
Charting for **Android**	Product Information Download
Charting for **Silverlight**	Product Information Download

Scheduling	Links
Scheduling for **WPF**	Product Information Download
Scheduling for **WinForms**	Product Information Download
Scheduling for **Java Swing**	Product Information Download
Scheduling for **JavaScript**	Product Information Download
Scheduling for **ASP.NET**	Product Information Download
Scheduling for **Xamarin**	Product Information Download
Scheduling for **Silverlight**	Product Information Download

Spreadsheet	Links
Spreadsheet for **WPF**	Product Information Download
Spreadsheet for **WinForms**	Product Information Download

Spreadsheet for **Java Swing**	Product Information Download

Reporting	Links
Reporting for **WPF**	Product Information Download
Reporting for **WinForms**	Product Information Download
Reporting for **Silverlight**	Product Information Download

Virtual Keyboard	Links
Virtual Keyboard for **WPF**	Product Information Download
Virtual Keyboard for **WinForms**	Product Information Download
Virtual Keyboard for **Java Swing**	Product Information Download
Virtual Keyboard for **JavaScript**	Product Information Download

Mapping	Links
Mapping for **WPF**	Product Information Download
Mapping for **WinForms**	Product Information Download
Mapping for **ASP.NET**	Product Information Download
Mapping for **Silverlight**	Product Information Download

Suites	Links
Pack for **WPF**	Product Information Download
Pack for **WinForms**	Product Information Download
Pack for **Java Swing**	Product Information Download
Pack for **JavaScript**	Product Information Download
Pack for **ASP.NET**	Product Information Download
Pack for **Xamarin**	Product Information Download
Pack for **Silverlight**	Product Information Download
Pack for **Diagramming**	Product Information Download
Pack for **Scheduling**	Product Information Download

Acknowledgements

Our thanks to:

- Joe Stegman, who permitted us to use his PNGEncoder class in the MvcDiagram implementation.
- Jon Harris, who permitted us to port his PQTree Java code for the MindFusion.GraphLib implementation.
- Nader Alkhatib, who contributed his LogicGates project as a sample for the MvcDiagram package.
- Mark James for the Silk icon set (http://www.famfamfam.com/lab/icons/silk/)
- ic#code for their #ziplib, used for implementing some custom MSI actions in MindFusion setup kits.