RibbonPanel

Source, CustomPanelBackground, CustomSlideOutPanelBackground, CustomPanelTitleBarBackground, HighlightPanelTitleBar, HighlightWhenCollapsed, IsVisible, IsEnabled, ResizeStyle, FloatingOrientation, CanToggleOrientation, Id, IsContextualTabThemeIgnored

```
<RibbonPanel CustomPanelBackground="" CustomSlideOutPanelBackground=""
CustomPanelTitleBarBackground="" HighlightPanelTitleBar="False"
HighlightWhenCollapsed="False" IsVisible="True" IsEnabled="True"
ResizeStyle="None" FloatingOrientation="Vertical"
CanToggleOrientation="True" Id=""
IsContextualTabThemeIgnored="False">...</RibbonPanel>
```

```
// Gets or sets the source that contains the ribbon items to be displayed by
this panel. The default value is null.
Source = null
CustomPanelBackground = null
CustomSlideOutPanelBackground = null
CustomPanelTitleBarBackground = null
// Accesses the highlight state for the ribbon panel's title bar.
HighlightPanelTitleBar = False
// Accesses the highlight state for the ribbon panel's title bar.
HighlightWhenCollapsed = False
// If the value is true, the panel is enabled. If the value is false, the
panel is disabled. When a panel is disabled all the items in the panel are
disabled. The default value is true. To disable all panels in a tab use
RibbonTab.IsPanelEnabled. If the value is true, the panel is visible in the
ribbon. If the value is false, it is hidden in the ribbon. Both visible and
hidden panels of a tab are available in the ribbon's right-click menu under
the Panels menu option, which allows the user to show or hide the panels. If
the panel's IsAnonymous property is set to false, it is not included in the
right-click menu and the user cannot control its visibility. The default
value is true.
IsVisible = True
// Gets or sets the value that indicates whether this panel is enabled. If
the value is true, the panel is enabled. If the value is false, the panel is
disabled. When a panel is disabled all the items in the panel are disabled.
The default value is true. To disable all panels in a tab use
RibbonTab.IsPanelEnabled.
IsEnabled = True
// Enum:
[None,NeverResizeItemWidth,NeverHideText,NeverChangeImageSize,NeverCollapse
Panel,NeverTrimTitle,NeverCollapseItem]
ResizeStyle = RibbonResizeStyles.None
// Gets or sets the orientation to be used when the panel is floating. This
property is applicable only when the panel is floating. The orientation of a
floating panel can be horizontal or vertical. The orientation can be switched
by the user with the Orientation button in the panel frame. Set the
CanToggleOrientation property to false to hide the Orientation button and
hinder the user from changing the orientation.
// Enum: [Horizontal,Vertical]
FloatingOrientation = Orientation.Vertical
CanToggleOrientation = True
Id = ""
IsContextualTabThemeIgnored = False
```

RibbonPanelSource

Id, Title, Name, Description, Tag, DialogLauncher, KeyTip, IsSlideOutPanelVisible

```
<RibbonPanelSource Id="" Title="" Name="" Description="" Tag="" KeyTip=""
IsSlideOutPanelVisible="True">
    <DialogLauncher Orientation="Horizontal" CommandHandler=""
IsCheckable="False" IsActive="False" Id="" Text="" LargeImage="" KeyTip=""
Image="" Tag="" Name="" GroupName="" IsToolTipEnabled="True" ToolTip=""
ShowImage="True" MinWidth="0" Width="NaN" ResizeStyle="ResizeWidth"
Size="Standard" ShowText="False" IsVisible="True" IsEnabled="True"
Description="" HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True" />
    ...
</RibbonPanelSource>
```

```
// Gets or sets the id for the panel source. The framework does not use or
validate this id. It is left to an application to set and use this id. The
default value is null.
Id = null
// The panel title set with this property is displayed in the panel's title
bar in the ribbon. The default value is null.
Title = null
// Gets or sets the name of the ribbon panel. The framework uses the Title
property of the panel to display the panel title in the ribbon. The name
property is not currently used by the framework. Applications can use this
property to store a longer name for a panel if this is required in other UI
customization dialogs. The default value is null.
Name = null
// Gets or sets the panel description text. The description text is not
currently used by the framework. Applications can use this to store a
description if it is required in other UI customization dialogs. The default
value is null.
Description = null
// Gets or sets the custom data object in the panel source. This property
can be used to store any object a as custom data object in a panel source.
This data is not used by the framework. The default value is null.
Tag = null
// Gets or sets the command item to be used as the panel's dialog launcher.
The dialog launcher is displayed as a small button in the panel title bar.
Clicking the button raises a command that follows the standard ribbon command
routing. If this property is null the panel does not have a dialog launcher
button. The default value is null.
DialogLauncher = null
// Gets or sets the name of the ribbon panel. The framework uses the Title
property of the panel to display the panel title in the ribbon. The name
property is not currently used by the framework. Applications can use this
property to store a longer name for a panel if this is required in other UI
customization dialogs. The default value is null.
KeyTip = null
IsSlideOutPanelVisible = True
```

RibbonSubPanelSource

Id, Description, Name, Tag

```
<RibbonSubPanelSource
Id="" Description="" Name="" Tag="">...</RibbonSubPanelSource>
```

```
Id = null
Description = null
Name = null
Tag = null
```

RibbonTab

Title, Name, Description, Id, IsVisible, IsActive, IsEnabled, IsPanelEnabled, IsContextualTab, IsMergedContextualTab, AllowTearOffContextualPanels, KeyTip, IsVisited, Tag

```
<RibbonTab Title="" Name="" Description="" Id="" IsVisible="True"
IsActive="False" IsEnabled="True" IsPanelEnabled="True"
IsContextualTab="False" IsMergedContextualTab="False"
AllowTearOffContextualPanels="False" KeyTip="" IsVisited="False"
Tag="">...</RibbonTab>
```

```
// Gets or sets the tab title. The title set with this property is displayed
in the tab button for this tab in the ribbon. The default value is null.
Title = null
// Gets or sets the name of the ribbon tab. The framework uses the Title
property of the tab to display the tab title in the ribbon. The name property
is not currently used by the framework. Applications can use this property
to store a longer name for the tab if it is required in other UI customization
dialogs. The default value is null.
Name = null
// Gets or sets a description text for the tab. The description text is not
currently used by the framework. Applications can use this to store a
description if it is required in other UI customization dialogs. The default
value is null.
Description = null
// Gets or sets the id for the tab. The framework does not use or validate
this id. It is left to the applications to set this id and use it. The default
value is null.
Id = null
// Gets or sets the value that indicates whether the tab is visible in the
ribbon. If the value is true, the tab is visible in the ribbon. If the value
is false, it is hidden in ribbon. Both visible and hidden tabs are available
in the ribbon by right-clicking the menu under the Tabs menu option, which
allows the user to show or hide the tabs. If the tab's IsAnonymous property
is set to false, it is not included in the right-click menu, and the user
cannot control its visibility. If an active tab is hidden, the next or
previous visible tab is set as the active tab. The default value is true.
IsVisible = True
// Gets or sets the value that indicates whether this tab is the active tab.
Hidden tabs and merged contextual tabs cannot be the active tab. Setting this
property to true for such tabs will fail, and no exception will be thrown.
IsActive = False
IsEnabled = True
IsPanelEnabled = True
```

```
// Assesses whether the tab is regular tab or contextual tab. If it is true
the tab is contextual tab, and false if it is regular tab. This is a dependency
property registered with WPF. Please see the Microsoft API for more
information. The default value is false.
IsContextualTab = False
IsMergedContextualTab = False
AllowTearOffContextualPanels = False
// Gets or sets the keytip for the tab. Keytips are displayed in the ribbon
when navigating the ribbon with the keyboard. If this property is null or
empty, the keytip will not appear for this tab, and the tab will not support
activation through the keyboard. The default value is null.
KeyTip = null
// Gets or sets the value that indicates whether the tab is visible in the
ribbon. If the value is true, the tab is visible in the ribbon. If the value
is false, it is hidden in ribbon. Both visible and hidden tabs are available
in the ribbon by right-clicking the menu under the Tabs menu option, which
allows the user to show or hide the tabs. If the tab's IsAnonymous property
is set to false, it is not included in the right-click menu, and the user
cannot control its visibility. If an active tab is hidden, the next or
previous visible tab is set as the active tab. The default value is true.
IsVisited = False
// Gets or sets custom data object in the tab. This property can be used to
store any object as a custom data object in a tab. This data is not used by
the framework. The default value is null.
Tag = null
```

RibbonRowBreak : RibbonItem

Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonRowBreak Id="" Text="" LargeImage="" KeyTip="" Image="" Tag=""
Name="" GroupName="" IsToolTipEnabled="True" ToolTip="" ShowImage="True"
MinWidth="100" Width="200" ResizeStyle="ResizeWidth" Size="Standard"
ShowText="False" IsVisible="True" IsEnabled="True" Description=""
HelpTopic="" ShowToolTipOnDisabled="True" AllowInToolBar="True" />
```

RibbonRowPanel : RibbonItem

Source, ResizePriority, SubPanelResizeStyle, IsTopJustified, AreItemsArrangedFromRightToLeft, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonRowPanel ResizePriority="100" SubPanelResizeStyle="None"
IsTopJustified="False" AreItemsArrangedFromRightToLeft="False" Id=""
Text="" LargeImage="" KeyTip="" Image="" Tag="" Name="" GroupName=""
IsToolTipEnabled="True" ToolTip="" ShowImage="True" MinWidth="100"
Width="200" ResizeStyle="ResizeWidth" Size="Standard" ShowText="False"
IsVisible="True" IsEnabled="True" Description="" HelpTopic=""
ShowToolTipOnDisabled="True" AllowInToolBar="True">...</RibbonRowPanel>
```

```
Source = null
ResizePriority = 100
// Enum: [None,NeverHideText,NeverWrap,NeverShrink,NoResize]
SubPanelResizeStyle = RibbonRowPanelResizeStyle.None
IsTopJustified = False
AreItemsArrangedFromRightToLeft = False
```

(abstract) RibbonItem

Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonItem Id="" Text="" LargeImage="" KeyTip="" Image="" Tag="" Name=""
GroupName="" IsToolTipEnabled="True" ToolTip="" ShowImage="True"
MinWidth="100" Width="200" ResizeStyle="ResizeWidth" Size="Standard"
ShowText="False" IsVisible="True" IsEnabled="True" Description=""
HelpTopic="" ShowToolTipOnDisabled="True" AllowInToolBar="True" />
```

```
// Gets or sets the item id. This id is used as the automation id for the
corresponding control in the UI. The framework does not otherwise use or
validate it. It is up to the application to set and use this id. The default
value is null.
Id = ""
// Gets or sets the text of a ribbon item. There are two properties that
support text content in ribbon items: Text and Name. The Text property is
meant for short text and is used wherever space is premium (for example, text
that appears directly in ribbon face). The Name property is meant for longer
text and is used wherever space is not an issue (for example, text that
appears in a drop-down list, tooltip, etc.). Both properties can be set for
a ribbon item. The visibility of text in the ribbon is controlled by the
ShowText property, which must be true to make the text visible. Text may also
be suppressed in one or more panels in a horizontal ribbon when there is not
enough space to display all the panels. The default value is null.
Text = null
// Gets or sets the image to be used when the item is displayed in large
size.Many ribbon items can appear in two sizes: Standard and Large. Some
ribbon items support only standard size (example: RibbonCombo) and some
support both (example: RibbonButton). Ribbon items also support ico files
containing multiple images in different sizes (only the first two images will
be used). If the ico file contains multiple size images, use it to set only
the Image or LargeImage property. The visibility of the image in a ribbon is
controlled by the ShowImage property, which must be set to true to make the
image visible.
LargeImage = null
// Gets or sets the keytip for the item. Keytips are displayed in the ribbon
and are used to navigate the ribbon through the keyboard. This property must
be set for a ribbon item to support a keytip. If this property is not set, a
keytip will not appear for this item, and the item will not support activation
through the keyboard. The default value is null.
KeyTip = null
```

```
// Gets or sets the image to be used when the item is displayed in standard
size. Many ribbon items can appear in two sizes: Standard and Large. Some
ribbon items support only standard size (example: RibbonCombo) and some
support both (example: RibbonButton). Ribbon items also support ico files
containing multiple images in different sizes (only the first two images will
be used). If the ico file contains multiple size images, use it to set only
the Image or LargeImage property. The visibility of the image in a ribbon is
controlled by the ShowImage property, which must be set to true to make the
image visible.
Image = null
// Gets or sets custom data object in the tab. This property can be used to
store any object as a custom data object in a tab. This data is not used by
the framework. The default value is null.
Tag = null
// Gets or sets the name of a ribbon item. There are two properties that
support text content in ribbon items: Text and Name. The Text property is
meant for short text and is used wherever space is premium (for example, text
that appears directly in ribbon face). The Name property is meant for longer
text and is used wherever space is not an issue (for example, text that
appears in a drop-down list, tooltip, etc.). Both properties can be set for
a ribbon item. The visibility of text in the ribbon is controlled by the
ShowText property, which must be true to make the text visible. Text may also
be suppressed in one or more panels in a horizontal ribbon when there is not
enough space to display all the panels. The default value is null.
Name = null
// Gets or sets the description text for the ribbon item. The description
text is used in the application menu, tooltips, and drop-down lists in a
RibbonListButton when the list style is set to Descriptive.
GroupName = null
// Gets or sets the value indicating whether F1 help is enabled in the item's
tooltip. If this value is true, F1 help is enabled in the item's tooltip;
otherwise, F1 help is disabled in the item' s tooltip. The default value is
true.
IsToolTipEnabled = True
// Gets or sets an object to be used as the tooltip for the item. If the
object is a string, it displays the string as a tooltip without any
formatting. If the object is of any other type, and the object has a data
template, the tooltip will use the data template to display the object. If
this property is null, a default tooltip of type RibbonToolTip is created at
runtime using the Name, Text, and Description properties of the item. If this
property is set to an empty string, the tooltip will be suppressed, and the
item will not show any tooltip. ToolTip can also be suppressed by setting
the IsToolTipEnabled property to false. The default value is null.
ToolTip = null
// Gets or sets the value indicating whether the item image is visible. If
the value is true, the image is visible in the ribbon, provided that a valid
image has been set with the Image property. If the value is false, the image
is not visible. The default value is true. Derived classes may override this
default.
ShowImage = True
// Gets or sets the minimum width of the item. This property is used only by
resizeable items. Resizeable items are items whose width is resized to fit
in the space available. For example, RibbonCombo and RibbonGallery are
resizeable items, and RibbonButton is not a resizeable item. If there is
enough space, resizeable items are displayed in full width, which is set with
the Width property. When there is not enough space, the item will be resized
to fit in the available space. The width of the item will not go below
MinWidth. The value for this property must be zero or positive and cannot be
Infinity. The value must be in device-independent units. The default value
is 100.
MinWidth = 100
```

```
// Gets or sets the width of the item. This property is used only by resizeable
items. Resizeable items are items whose width is resized to fit in the space
available. For example, RibbonCombo and RibbonGallery are resizeable items,
and RibbonButton is not a resizeable item. If there is enough space,
resizeable items are displayed in full width. When there is not enough space,
the item will be resized to fit in the available space. The width of the item
will not go below MinWidth. The value for this property must be zero or
positive and cannot be Infinity. The value must be in device-independent
units. The default value is 200.
Width = 200
// Enum: [NoResize,ChangeSize,HideText,ResizeWidth,ResizeHeight,Collapse]
ResizeStyle = RibbonItemResizeStyles.ResizeWidth

// Gets or sets the size with which the item is displayed. This property is
supported only by RibbonButton and RibbonLabel as well as classes derived
from them. Other ribbon items will ignore this property. The default value
is Standard.
// Enum: [Standard,Large]
Size = RibbonItemSize.Standard
// Gets or sets the value indicating whether the item text is visible. If
the value is true, the text for the item is visible in the ribbon, provided
that a valid text has been set for this item with the Text property. If the
value is false, the text is not visible in the ribbon. The default value is
false. Derived classes may override this default.
ShowText = False
// Gets or sets the value to indicating whether the item is visible in the
ribbon. If this property is true, the item is visible in the ribbon;
otherwise, the item is hidden in ribbon and does not occupy any space. The
default value is true.
IsVisible = True
// Gets or sets the value indicating whether the item is enabled in the
ribbon.If this property is true, the item is enabled in the ribbon; otherwise,
it is disabled. Disabled items do not respond any interaction. The default
value is true.
IsEnabled = True
// Gets or sets the description text for the ribbon item. The description
text is used in the application menu, tooltips, and drop-down lists in a
RibbonListButton when the list style is set to Descriptive.
Description = null
HelpTopic = null
ShowToolTipOnDisabled = True
AllowInToolBar = True
```

(abstract) RibbonCommandItem : RibbonItem

CommandHandler, IsCheckable, IsActive, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonCommandItem CommandHandler="" IsCheckable="False" IsActive="False"
Id="" Text="" LargeImage="" KeyTip="" Image="" Tag="" Name="" GroupName=""
IsToolTipEnabled="True" ToolTip="" ShowImage="True" MinWidth="100"
Width="200" ResizeStyle="ResizeWidth" Size="Standard" ShowText="False"
IsVisible="True" IsEnabled="True" Description="" HelpTopic=""
ShowToolTipOnDisabled="True" AllowInToolBar="True" />
```

```
CommandHandler = null
```

```
// Gets or sets the value that indicates if this is a checkable item. This
property is used only by item types that support the toggling of items. For
example, items in a drop down list of a RibbonMenuItem or
RibbonChecklistButton use this property. The default value is false.
IsCheckable = False
IsActive = False
```

RibbonPanelBreak : RibbonItem

Id, **Text**, **LargeImage**, **KeyTip**, **Image**, **Tag**, **Name**, **GroupName**, **IsToolTipEnabled**, **ToolTip**, **ShowImage**, **MinWidth**, **Width**, **ResizeStyle**, **Size**, **ShowText**, **IsVisible**, **IsEnabled**, **Description**, **HelpTopic**, **ShowToolTipOnDisabled**, **AllowInToolBar**

```
<RibbonPanelBreak Id="" Text="" LargeImage="" KeyTip="" Image="" Tag=""
Name="" GroupName="" IsToolTipEnabled="True" ToolTip="" ShowImage="True"
MinWidth="100" Width="200" ResizeStyle="ResizeWidth" Size="Standard"
ShowText="False" IsVisible="True" IsEnabled="True" Description=""
HelpTopic="" ShowToolTipOnDisabled="True" AllowInToolBar="True" />
```

RibbonSlider : RibbonItem

Minimum, **Maximum**, **TextBox1Visibility**, **TextBox1Editable**, **IsSnapToTickEnabled**, **TextBox1Width**, **TickPlacement**, **Value**, **Ticks**, **TextBox1Text**, **Id**, **Text**, **LargeImage**, **KeyTip**, **Image**, **Tag**, **Name**, **GroupName**, **IsToolTipEnabled**, **ToolTip**, **ShowImage**, **MinWidth**, **Width**, **ResizeStyle**, **Size**, **ShowText**, **IsVisible**, **IsEnabled**, **Description**, **HelpTopic**, **ShowToolTipOnDisabled**, **AllowInToolBar**

```
<RibbonSlider Minimum="0" Maximum="0" TextBox1Visibility="Collapsed"
TextBox1Editable="False" IsSnapToTickEnabled="True" TextBox1Width="27"
TickPlacement="None" Value="0" Ticks="" TextBox1Text="" Id="" Text=""
LargeImage="" KeyTip="" Image="" Tag="" Name="" GroupName=""
IsToolTipEnabled="True" ToolTip="" ShowImage="True" MinWidth="100"
Width="200" ResizeStyle="ResizeWidth" Size="Standard" ShowText="False"
IsVisible="True" IsEnabled="True" Description="" HelpTopic=""
ShowToolTipOnDisabled="True" AllowInToolBar="True" />
```

```
Minimum = 0
Maximum = 0
// Enum: [Visible,Hidden,Collapsed]
TextBox1Visibility = Visibility.Collapsed
TextBox1Editable = False
IsSnapToTickEnabled = True
TextBox1Width = 27
// Enum: [None,TopLeft,BottomRight,Both]
TickPlacement = TickPlacement.None
Value = 0
Ticks = null
TextBox1Text = null
```

(abstract) **RibbonList** : RibbonItem

IsGrouping, **DropDownWidth**, **MaxDropDownHeight**, **Id**, **Text**, **LargeImage**, **KeyTip**, **Image**, **Tag**, **Name**, **GroupName**, **IsToolTipEnabled**, **ToolTip**, **ShowImage**, **MinWidth**, **Width**, **ResizeStyle**, **Size**, **ShowText**, **IsVisible**, **IsEnabled**, **Description**, **HelpTopic**, **ShowToolTipOnDisabled**, **AllowInToolBar**

```
<RibbonList IsGrouping="False" DropDownWidth="NaN" MaxDropDownHeight="NaN"
Id="" Text="" LargeImage="" KeyTip="" Image="" Tag="" Name="" GroupName=""
IsToolTipEnabled="True" ToolTip="" ShowImage="False" MinWidth="100"
Width="200" ResizeStyle="ResizeWidth" Size="Standard" ShowText="False"
IsVisible="True" IsEnabled="True" Description="" HelpTopic=""
ShowToolTipOnDisabled="True" AllowInToolBar="True" />...</RibbonList>
```

```
// Gets or sets the value that indicates whether the drop-down list should support the grouping of items. Only RibbonCombo supports grouping. RibbonGallery does not support grouping. Grouping is done using the property RibbonItem.GroupName in the drop-down items. If this property is true, grouping is enabled in the drop-down list. If it is false, grouping is not enabled. The default value is false.
IsGrouping = False
// Gets or sets the width of the drop-down window that is displayed when a drop-down item is opened. The width must be in device independent units. The default value is NaN. The minimum drop-down width is equal to the control width. Thus, if the value set in this property is less than the control width, the value is ignored.
DropDownWidth = double.NaN
// Gets or sets the maximum height of the drop-down window that is displayed when a drop-down item is opened. The height must be in device independent units. The actual drop-down height depends on the number of items in the list and will not exceed the value set in this property. The default value is a calculated value that is based on system max screen height parameters.
MaxDropDownHeight = double.NaN
```

RibbonLabel : RibbonItem

Id, **Text**, **LargeImage**, **KeyTip**, **Image**, **Tag**, **Name**, **GroupName**, **IsToolTipEnabled**, **ToolTip**, **ShowImage**, **MinWidth**, **Width**, **ResizeStyle**, **Size**, **ShowText**, **IsVisible**, **IsEnabled**, **Description**, **HelpTopic**, **ShowToolTipOnDisabled**, **AllowInToolBar**

```
<RibbonLabel Id="" Text="" LargeImage="" KeyTip="" Image="" Tag="" Name=""
GroupName="" IsToolTipEnabled="False" ToolTip="" ShowImage="True"
MinWidth="0" Width="NaN" ResizeStyle="NoResize" Size="Standard"
ShowText="True" IsVisible="True" IsEnabled="True" Description=""
HelpTopic="" ShowToolTipOnDisabled="True" AllowInToolBar="True" />
```

RibbonSeparator : RibbonItem

SeparatorStyle, **Id**, **Text**, **LargeImage**, **KeyTip**, **Image**, **Tag**, **Name**, **GroupName**, **IsToolTipEnabled**, **ToolTip**, **ShowImage**, **MinWidth**, **Width**, **ResizeStyle**, **Size**, **ShowText**, **IsVisible**, **IsEnabled**, **Description**, **HelpTopic**, **ShowToolTipOnDisabled**, **AllowInToolBar**

```
<RibbonSeparator SeparatorStyle="Line" Id="" Text="" LargeImage=""
KeyTip="" Image="" Tag="" Name="" GroupName="" IsToolTipEnabled="True"
ToolTip="" ShowImage="True" MinWidth="0" Width="NaN" ResizeStyle="NoResize"
Size="Standard" ShowText="False" IsVisible="True" IsEnabled="True"
Description="" HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True" />
```

```
// Gets or sets the value specifying the separator style. Separator styles
are used to set the appearance of the separator in a ribbon.
// Enum: [Line,Line,Spacer,Invisible]
SeparatorStyle = RibbonSeparatorStyle.Line
```

RibbonSpinner : RibbonItem

Value, **Change**, **Maximum**, **Minimum**, **IsEditable**, **ResizableBoxWidth**, **Id**, **Text**, **LargeImage**, **KeyTip**, **Image**, **Tag**, **Name**, **GroupName**, **IsToolTipEnabled**, **ToolTip**, **ShowImage**, **MinWidth**, **Width**, **ResizeStyle**, **Size**, **ShowText**, **IsVisible**, **IsEnabled**, **Description**, **HelpTopic**, **ShowToolTipOnDisabled**, **AllowInToolBar**

```
<RibbonSpinner Value="" Change="" Maximum="" Minimum="" IsEditable="False"
ResizableBoxWidth="NaN" Id="" Text="" LargeImage="" KeyTip="" Image=""
Tag="" Name="" GroupName="" IsToolTipEnabled="True" ToolTip=""
ShowImage="False" MinWidth="100" Width="200" ResizeStyle="ResizeWidth"
Size="Standard" ShowText="False" IsVisible="True" IsEnabled="True"
Description="" HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True" />
```

```
// Gets or sets the current value. The data types int and double are supported
by default. To support other data types, derive from this class and override
the virtual methods. The default value is null.
Value = null
// Gets or sets the value specifying the amount of change that occurs when
the up or down spin button is pressed. The data type of the value assigned
to this property must be same as the data type of the Value property. The
data types int and double are supported by default. To support other data
types, derive from this class and override the virtual methods. The default
value is null.
Change = null
// Gets or sets the maximum value of the spin range. The data type of the
value assigned to this property must be same as the data type of the Value
property. The data types int and double are supported by default. To support
other data types, derive from this class and override the virtual methods.
The default value is null.
Maximum = null
```

```
// Gets or sets the minimum value of the spin range. The data type of the value assigned to this property must be same as the data type of the Value property. The data types int and double are supported by default. To support other data types, derive from this class and override the virtual methods. The default value is null.The default value is null.
Minimum = null
// Gets or sets the value that indicates whether the value is directly editable. If the value is true, in addition to changing the value with spin buttons, the value can be entered directly in the edit control of the spinner. If it is false, the spinner value can only be changed using the spin buttons. The default value is false.
IsEditable = False
ResizableBoxWidth = double.NaN
```

RibbonTextBox : RibbonItem

Value, **ImageLocation**, **ShowImageAsButton**, **SelectTextOnFocus**, **AcceptTextOnLostFocus**, **InvokesCommand**, **Prompt**, **CommandHandler**, **IsEmptyTextValid**, **ResizableBoxWidth**, **Id**, **Text**, **LargeImage**, **KeyTip**, **Image**, **Tag**, **Name**, **GroupName**, **IsToolTipEnabled**, **ToolTip**, **ShowImage**, **MinWidth**, **Width**, **ResizeStyle**, **Size**, **ShowText**, **IsVisible**, **IsEnabled**, **Description**, **HelpTopic**, **ShowToolTipOnDisabled**, **AllowInToolBar**

```
<RibbonTextBox Value="" ImageLocation="Left" ShowImageAsButton="False"
SelectTextOnFocus="False" AcceptTextOnLostFocus="True"
InvokesCommand="False" Prompt="" CommandHandler="" IsEmptyTextValid="True"
ResizableBoxWidth="NaN" Id="" Text="" LargeImage="" KeyTip="" Image=""
Tag="" Name="" GroupName="" IsToolTipEnabled="True" ToolTip=""
ShowImage="False" MinWidth="100" Width="200" ResizeStyle="ResizeWidth"
Size="Standard" ShowText="False" IsVisible="True" IsEnabled="True"
Description="" HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True" />
```

```
// Gets or sets the Value property. The value can be a string or other data type. If it is not a string, you must derive from this class and implement the virtual data conversion methods. The default value is null.
Value = null
// Gets or sets the location for the image that is displayed in the text box. ShowImage must be set to true to see the image. This property is ignored if ShowImage is false. The default value is Left.
// Enum: [Left,InsideLeft,InsideRight]
ImageLocation = RibbonTextBoxImageLocation.Left
// Gets or sets the value that indicates whether the Image set in the text box should be displayed as a clickable button. If ShowImageAsButton is true, ShowImage must also be true, and ImageLocation must be set to InsideLeft or InsideRight. If ShowImage is false, the button will not be visible. If ImageLocation is Left, this property will not have any effect. Clicking this button will invoke the command handler if InvokesCommand is true. The default value is false.
ShowImageAsButton = False
// Gets or sets the value that indicates whether the text is selected when the text box gains focus. If the value is true, all the text in the text box is selected when the text box gets keyboard focus. If it is false, the text is not selected. The default value is false.
SelectTextOnFocus = False
```

```
// Gets or sets the value that indicates whether edited text should be
accepted when the text box loses focus before enter is pressed or the text
box button is clicked. This property controls whether the edited text is
accepted or rejected if the text box loses focus before the text box button
is clicked or the enter key pressed. If the value is true, the edited text
is accepted. If it is false, the edited text is rejected and the previous
value is restored. If the text is accepted, the command handler is invoked
if InvokesCommand is true. If the text box button is clicked or the enter
key is pressed, the text is accepted regardless of this property value. The
default value is true.
AcceptTextOnLostFocus = True
// Gets or sets the value that indicates whether the command handler needs
to be invoked whenever text is changed.  If the value is true, the command
handler is invoked when text is changed in the UI; if the value is false, no
command is invoked. The default value is false.
InvokesCommand = False
// Gets or sets the prompt text for the text box. Prompt text is displayed
when the text box is empty and does not have keyboard focus. The default
value is null.
Prompt = null
// Gets or sets the command handler to be called when the text is changed.The
property InvokesCommand must be true for the command handler to be called.
If it is false, no command will be invoked by the text box. Also, the command
will not be invoked by the text box if the text is not accepted (i.e.,
validation has failed). The default value is null.
CommandHandler = null
// Gets or sets the value that indicates whether empty text should be
considered a valid value. If IsEmptyTextValid is true, empty text is
considered valid text, and the command is invoked even if the text is empty;
if it is false, when the text is empty, the command is not invoked, and the
text box button is disabled.
IsEmptyTextValid = True
ResizableBoxWidth = double.NaN
```

ProgressBarSource : RibbonCommandItem **(AutoCAD)**

HasCancelButton, **CurrentOperation**, **MaximumValue**, **MinimumValue**, **CurrentValue**, **CommandHandler**, **IsCheckable**, **IsActive**, **Id**, **Text**, **LargeImage**, **KeyTip**, **Image**, **Tag**, **Name**, **GroupName**, **IsToolTipEnabled**, **ToolTip**, **ShowImage**, **MinWidth**, **Width**, **ResizeStyle**, **Size**, **ShowText**, **IsVisible**, **IsEnabled**, **Description**, **HelpTopic**, **ShowToolTipOnDisabled**, **AllowInToolBar**

```
<ProgressBarSource HasCancelButton="False" CurrentOperation=""
MaximumValue="100" MinimumValue="0" CurrentValue="0" CommandHandler=""
IsCheckable="False" IsActive="False" Id="" Text="" LargeImage="" KeyTip=""
Image="" Tag="" Name="" GroupName="" IsToolTipEnabled="True" ToolTip=""
ShowImage="True" MinWidth="100" Width="200" ResizeStyle="ResizeWidth"
Size="Standard" ShowText="False" IsVisible="True" IsEnabled="True"
Description="" HelpTopic="" ShowToolTipOnDisabled="True" />
```

```
AllowInToolBar = True
HasCancelButton = False
CurrentOperation = ""
MaximumValue = 100
MinimumValue = 0
CurrentValue = 0
```

RibbonCheckBox : RibbonCommandItem

```
CommandHandler, IsCheckable, IsActive, Id, Text, LargeImage, KeyTip, Image,
Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth,
Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description,
HelpTopic, ShowToolTipOnDisabled, AllowInToolBar
```

```
<RibbonCheckBox CommandHandler="" IsCheckable="False" IsActive="False"
Id="" Text="" LargeImage="" KeyTip="" Image="" Tag="" Name="" GroupName=""
IsToolTipEnabled="True" ToolTip="" ShowImage="True" MinWidth="100"
Width="200" ResizeStyle="ResizeWidth" Size="Standard" ShowText="False"
IsVisible="True" IsEnabled="True" Description="" HelpTopic=""
ShowToolTipOnDisabled="True" AllowInToolBar="True" />
```

(abstract) **RibbonListButton** : RibbonButton

```
IsSplit, IsGrouping, AllowOrientation, Orientation, CommandHandler,
IsCheckable, IsActive, Id, Text, LargeImage, KeyTip, Image, Tag, Name,
GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width,
ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic,
ShowToolTipOnDisabled, AllowInToolBar
```

```
<RibbonListButton IsSplit="True" IsGrouping="False"
AllowOrientation="False" Orientation="Horizontal" CommandHandler=""
IsCheckable="False" IsActive="False" Id="" Text="" LargeImage="" KeyTip=""
Image="" Tag="" Name="" GroupName="" IsToolTipEnabled="True" ToolTip=""
ShowImage="True" MinWidth="0" Width="NaN" ResizeStyle="ResizeWidth"
Size="Standard" ShowText="False" IsVisible="True" IsEnabled="True"
Description="" HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True">...</RibbonListButton>
```

```
// Gets or sets the value that indicates whether the list button is to behave
like a split button. If this property is true, the list button supports
executing the button without opening the drop-down list, and the drop-down
list is opened by clicking the arrow. If it is false, the list button always
opens the drop-down list when clicked, and items need to be executed from
the drop-down list. The default value is true.
IsSplit = True
// Gets or sets the value that indicates whether the drop-down list supports
the grouping of items. Grouping is accomplished by setting the property
RibbonItem.GroupName for the drop-down items, so the items in the drop-down
list should set the group name with that property. If this property is true,
grouping is enabled in the drop-down list. If it is false, grouping is not
enabled. The default value is false.
IsGrouping = False
AllowOrientation = False
```

RibbonButton : RibbonCommandItem

Orientation, CommandHandler, IsCheckable, IsActive, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonButton Orientation="Horizontal" CommandHandler=""
IsCheckable="False" IsActive="False" Id="" Text="" LargeImage="" KeyTip=""
Image="" Tag="" Name="" GroupName="" IsToolTipEnabled="True" ToolTip=""
ShowImage="True" MinWidth="0" Width="NaN" ResizeStyle="ResizeWidth"
Size="Standard" ShowText="False" IsVisible="True" IsEnabled="True"
Description="" HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True" />
```

```
// Accesses the orientation of text and image. This is a dependency property.
The default value is Horizontal.
// Enum: [Horizontal,Vertical]
Orientation = Orientation.Horizontal
```

RibbonMenuItem : RibbonCommandItem

CommandHandler, IsCheckable, IsActive, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonMenuItem CommandHandler="" IsCheckable="True" IsActive="False" Id=""
Text="" LargeImage="" KeyTip="" Image="" Tag="" Name="" GroupName=""
IsToolTipEnabled="True" ToolTip="" ShowImage="True" MinWidth="100"
Width="200" ResizeStyle="ResizeWidth" Size="Standard" ShowText="False"
IsVisible="True" IsEnabled="True" Description="" HelpTopic=""
ShowToolTipOnDisabled="True" AllowInToolBar="True">...</RibbonMenuItem>
```

RibbonToggleButton : RibbonCommandItem

CommandHandler, IsCheckable, IsActive, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonToggleButton CommandHandler="" IsCheckable="True" IsActive="False"
Id="" Text="" LargeImage="" KeyTip="" Image="" Tag="" Name="" GroupName=""
IsToolTipEnabled="True" ToolTip="" ShowImage="True" MinWidth="100"
Width="200" ResizeStyle="ResizeWidth" Size="Standard" ShowText="False"
IsVisible="True" IsEnabled="True" Description="" HelpTopic=""
ShowToolTipOnDisabled="True" AllowInToolBar="True" />
```

RibbonPanelSpacer : RibbonPanelSource

LeftBorderBrush, **RightBorderBrush**, **Id**, **Title**, **Name**, **Description**, **Tag**, **DialogLauncher**, **KeyTip**, **IsSlideOutPanelVisible**

```
<RibbonPanelSpacer Id="" Title="" Name="" Description="" Tag="" KeyTip=""
IsSlideOutPanelVisible="True" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <LeftBorderBrush>
        &lt;SolidColorBrush
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Color="#FFFF0000" /&gt;
    </LeftBorderBrush>
    <RightBorderBrush>
        &lt;SolidColorBrush
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Color="#FF0000FF" /&gt;
    </RightBorderBrush>
</RibbonPanelSpacer>
```

```
LeftBorderBrush = Transparent
RightBorderBrush = Transparent
```

RibbonFlowPanel : RibbonRowPanel

MaxRowNumber, **AreColumnsStatic**, **Source**, **ResizePriority**, **SubPanelResizeStyle**, **IsTopJustified**, **AreItemsArrangedFromRightToLeft**, **Id**, **Text**, **LargeImage**, **KeyTip**, **Image**, **Tag**, **Name**, **GroupName**, **IsToolTipEnabled**, **ToolTip**, **ShowImage**, **MinWidth**, **Width**, **ResizeStyle**, **Size**, **ShowText**, **IsVisible**, **IsEnabled**, **Description**, **HelpTopic**, **ShowToolTipOnDisabled**, **AllowInToolBar**

```
<RibbonFlowPanel MaxRowNumber="3" AreColumnsStatic="False"
ResizePriority="100" SubPanelResizeStyle="None" IsTopJustified="False"
AreItemsArrangedFromRightToLeft="False" Id="" Text="" LargeImage=""
KeyTip="" Image="" Tag="" Name="" GroupName="" IsToolTipEnabled="True"
ToolTip="" ShowImage="True" MinWidth="100" Width="200"
ResizeStyle="ResizeWidth" Size="Standard" ShowText="False" IsVisible="True"
IsEnabled="True" Description="" HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True">...</RibbonFlowPanel>
```

```
MaxRowNumber = 3
AreColumnsStatic = False
```

RibbonFoldPanel : RibbonRowPanel

SubPanelResizeStyle, DefaultSize, MaxSize, MinSize, Source, ResizePriority, SubPanelResizeStyle, IsTopJustified, AreItemsArrangedFromRightToLeft, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonFoldPanel SubPanelResizeStyle="None" DefaultSize="Medium"
MaxSize="Large" MinSize="Small" ResizePriority="100"
SubPanelResizeStyle="None" IsTopJustified="False"
AreItemsArrangedFromRightToLeft="False" Id="" Text="" LargeImage=""
KeyTip="" Image="" Tag="" Name="" GroupName="" IsToolTipEnabled="True"
ToolTip="" ShowImage="True" MinWidth="100" Width="200"
ResizeStyle="ResizeWidth" Size="Standard" ShowText="False" IsVisible="True"
IsEnabled="True" Description="" HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True">...</RibbonFoldPanel>
```

```
// Enum: [None,NoResize]
SubPanelResizeStyle = RibbonFoldPanelResizeStyle.None
// Enum: [Small,Medium,Large]
DefaultSize = RibbonFoldPanelSize.Medium
// Enum: [Small,Medium,Large]
MaxSize = RibbonFoldPanelSize.Large
// Enum: [Small,Medium,Large]
MinSize = RibbonFoldPanelSize.Small
```

RibbonCombo : RibbonList

TextPath, IsEditable, EditableText, IsTextSearchEnabled, CommandHandler, ResizableBoxWidth, IsGrouping, DropDownWidth, MaxDropDownHeight, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonCombo TextPath="Text" IsEditable="False" EditableText=""
IsTextSearchEnabled="True" CommandHandler="" ResizableBoxWidth="NaN"
IsGrouping="False" DropDownWidth="NaN" MaxDropDownHeight="NaN" Id=""
Text="" LargeImage="" KeyTip="" Image="" Tag="" Name="" GroupName=""
IsToolTipEnabled="True" ToolTip="" ShowImage="False" MinWidth="100"
Width="200" ResizeStyle="ResizeWidth" Size="Standard" ShowText="False"
IsVisible="True" IsEnabled="True" Description="" HelpTopic=""
ShowToolTipOnDisabled="True" AllowInToolBar="True">...</RibbonCombo>
```

```
TextPath = Text
// Gets or sets the value that indicates whether the combo box text is
editable. If this property is true, the combo box allows text to be entered
that is not in the list. The entered text is not added to the list. The
entered text can be validated in the EditableTextChanging event. The default
value is false.
IsEditable = False
```

```
// Gets or sets the editable text in the combo box. This property is applicable
only if IsEditable is true. The default value is null.
EditableText = null
IsTextSearchEnabled = True
// Gets or sets the command handler to be called when the RibbonCombo menu
items are executed. The command is routed to the first command handler found
while searching in the following order: 1. the command handler set in the
item. 2. the command handler set in the RibbonCombo. 3. the command handler
set in the root control that contains this item (ribbon, Quick Access Toolbar,
menu, or status bar). 4. the global command handler set in
ComponentManager.CommandHandler. The default value is null.
CommandHandler = null
ResizableBoxWidth = double.NaN
```

RibbonChecklistButton : RibbonListButton

IsSplit, **IsGrouping**, **AllowOrientation**, **Orientation**, **CommandHandler**, **IsCheckable**, **IsActive**, **Id**, **Text**, **LargeImage**, **KeyTip**, **Image**, **Tag**, **Name**, **GroupName**, **IsToolTipEnabled**, **ToolTip**, **ShowImage**, **MinWidth**, **Width**, **ResizeStyle**, **Size**, **ShowText**, **IsVisible**, **IsEnabled**, **Description**, **HelpTopic**, **ShowToolTipOnDisabled**, **AllowInToolBar**

```
<RibbonChecklistButton IsSplit="True" IsGrouping="False"
AllowOrientation="False" Orientation="Horizontal" CommandHandler=""
IsCheckable="False" IsActive="False" Id="" Text="" LargeImage="" KeyTip=""
Image="" Tag="" Name="" GroupName="" IsToolTipEnabled="True" ToolTip=""
ShowImage="True" MinWidth="0" Width="NaN" ResizeStyle="ResizeWidth"
Size="Standard" ShowText="False" IsVisible="True" IsEnabled="True"
Description="" HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True">...</RibbonChecklistButton>
```

RibbonMenuButton : RibbonListButton

IsSplit, **IsGrouping**, **AllowOrientation**, **Orientation**, **CommandHandler**, **IsCheckable**, **IsActive**, **Id**, **Text**, **LargeImage**, **KeyTip**, **Image**, **Tag**, **Name**, **GroupName**, **IsToolTipEnabled**, **ToolTip**, **ShowImage**, **MinWidth**, **Width**, **ResizeStyle**, **Size**, **ShowText**, **IsVisible**, **IsEnabled**, **Description**, **HelpTopic**, **ShowToolTipOnDisabled**, **AllowInToolBar**

```
<RibbonMenuButton IsSplit="True" IsGrouping="False"
AllowOrientation="False" Orientation="Horizontal" CommandHandler=""
IsCheckable="False" IsActive="False" Id="" Text="" LargeImage="" KeyTip=""
Image="" Tag="" Name="" GroupName="" IsToolTipEnabled="True" ToolTip=""
ShowImage="True" MinWidth="0" Width="NaN" ResizeStyle="ResizeWidth"
Size="Standard" ShowText="False" IsVisible="True" IsEnabled="True"
Description="" HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True">...</RibbonMenuButton>
```

RibbonRadioButtonGroup : RibbonListButton

MaxRow, MaxColumn, ExpandOrientation, CollapsedSize, CanCollapse, IsSplit, IsGrouping, AllowOrientation, Orientation, CommandHandler, IsCheckable, IsActive, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonRadioButtonGroup MaxRow="3" MaxColumn="10000"
ExpandOrientation="Horizontal" CollapsedSize="Standard" CanCollapse="True"
IsSplit="False" IsGrouping="False" AllowOrientation="False"
Orientation="Horizontal" CommandHandler="" IsCheckable="False"
IsActive="False" Id="" Text="" LargeImage="" KeyTip="" Image="" Tag=""
Name="" GroupName="" IsToolTipEnabled="True" ToolTip="" ShowImage="True"
MinWidth="0" Width="NaN" ResizeStyle="ResizeWidth" Size="Standard"
ShowText="False" IsVisible="True" IsEnabled="True" Description=""
HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True">...<RibbonRadioButtonGroup>
```

```
MaxRow = 3
MaxColumn = 10000
// Enum: [Horizontal,Vertical]
ExpandOrientation = Orientation.Horizontal
// Enum: [Standard,Large]
CollapsedSize = RibbonItemSize.Standard
CanCollapse = True
```

RibbonSplitButton : RibbonListButton

ListStyle, ListImageSize, IsSplit, IsGrouping, AllowOrientation, Orientation, CommandHandler, IsCheckable, IsActive, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonSplitButton ListStyle="List" ListImageSize="Large" IsSplit="True"
IsGrouping="False" AllowOrientation="False" Orientation="Horizontal"
CommandHandler="" IsCheckable="False" IsActive="False" Id="" Text=""
LargeImage="" KeyTip="" Image="" Tag="" Name="" GroupName=""
IsToolTipEnabled="True" ToolTip="" ShowImage="True" MinWidth="0"
Width="NaN" ResizeStyle="ResizeWidth" Size="Standard" ShowText="False"
IsVisible="True" IsEnabled="True" Description="" HelpTopic=""
ShowToolTipOnDisabled="True" AllowInToolBar="True">...</RibbonSplitButton>
```

```
// Enum: [Icon,List,Descriptive]
ListStyle = RibbonSplitButtonListStyle.List
// Enum: [Standard,Large]
ListImageSize = RibbonImageSize.Large
```

ApplicationMenuItem : RibbonMenuItem

IsSplit, SplitKeyTip, IsPinable, MaxDescriptionLines, CommandHandler, IsCheckable, IsActive, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<ApplicationMenuItem IsSplit="False" SplitKeyTip="" IsPinable="False"
MaxDescriptionLines="0" CommandHandler="" IsCheckable="False"
IsActive="False" Id="" Text="" LargeImage="" KeyTip="" Image="" Tag=""
Name="" GroupName="" IsToolTipEnabled="True" ToolTip="" ShowImage="True"
MinWidth="100" Width="200" ResizeStyle="ResizeWidth" Size="Standard"
ShowText="False" IsVisible="True" IsEnabled="True" Description=""
HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True">...<ApplicationMenuItem>
```

```
// Gets or sets the value indicating whether or not the menu item is to be
displayed as a split button type menu item. If the value is true the menu
item is displayed as a split button. Clicking the left portion of the split
button executes the item and clicking the right portion of the button opens
the next sub-level menu. This property is applicable only for main menu items
(first-level menu items) and is ignored when set in menu items in a second
or subsequent level. Setting this value to true does not make sense if the
menu item does not have a sub-menu. The default value is false.
IsSplit = False
// Gets or sets the keytip for the right portion of the split button menu
item which is used to open the next level menu. This property is applicable
only for split buttons (i.e. IsSplit=true). The default value is null.
SplitKeyTip = null
IsPinable = False
MaxDescriptionLines = 0
```

ToolBarShareButton : RibbonToggleButton **(>2025; ZW)**

CommandHandler, IsCheckable, IsActive, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, ResizeStyle, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<ToolBarShareButton CommandHandler="" IsCheckable="True" IsActive="False"
Id="" Text="" LargeImage="" KeyTip="" Image="" Tag="" Name="" GroupName=""
IsToolTipEnabled="True" ToolTip="" ShowImage="True" MinWidth="100"
Width="200" ResizeStyle="ResizeWidth" Size="Standard" ShowText="False"
IsVisible="True" IsEnabled="True" Description="" HelpTopic=""
ShowToolTipOnDisabled="True" AllowInToolBar="True" />
```

RibbonGallery : RibbonCombo

DisplayMode, ItemWidth, ItemHeight, TextPath, IsEditable, EditableText, IsTextSearchEnabled, CommandHandler, ResizableBoxWidth, IsGrouping, DropDownWidth, MaxDropDownHeight, Id, Text, LargeImage, KeyTip, Image, Tag, Name, GroupName, IsToolTipEnabled, ToolTip, ShowImage, MinWidth, Width, Size, ShowText, IsVisible, IsEnabled, Description, HelpTopic, ShowToolTipOnDisabled, AllowInToolBar

```
<RibbonGallery DisplayMode="" ItemWidth="NaN" ItemHeight="NaN"
TextPath="Text" IsEditable="False" EditableText=""
IsTextSearchEnabled="True" CommandHandler="" ResizableBoxWidth="NaN"
IsGrouping="False" DropDownWidth="NaN" MaxDropDownHeight="NaN" Id=""
Text="" LargeImage="" KeyTip="" Image="" Tag="" Name="" GroupName=""
IsToolTipEnabled="True" ToolTip="" ShowImage="False" MinWidth="100"
Width="200" Size="Standard" ShowText="False" IsVisible="True"
IsEnabled="True" Description="" HelpTopic="" ShowToolTipOnDisabled="True"
AllowInToolBar="True">...</RibbonGallery>
```

```
// Gets or sets the display mode of the gallery. The display mode is used to
specify the appearance of the gallery in the ribbon. The default value is
Window.
// Enum: [Window,ComboBox,LargeButton,StandardButton]
DisplayMode = GalleryDisplayMode.Window
ItemWidth = double.NaN
ItemHeight = double.NaN
```