INHALT

1. NEUE OBJEKTE IN SILVERLIGHT 2 2

1.1. BitmapImage 2

1.2. Border 2

1.3. ColumnDefinition 2

1.4. CornerRadius 2

1.5. DeepZoomImageTileSource 2

1.6. DiscreteObjectKeyFrame 2

1.7. Grid 2

1.8. GridLength 2

1.9. MouseButtonEventArgs 2

1.10. MultiScaleImage 2

1.11. MultiScaleSubImage 2

1.13. ObjectAnimationUsingKeyFrames 2

1.14. ObjectKeyFrameCollection 2

1.15. PasswordBox 2

1.16. Popup 2

1.17. RowDefinition 2

1.18. Setter 2

1.19. StackPanel 2

1.20. Style (Object) 2

1.21. TextBox 2

1.22. Thickness 2

2. NEUE METHODEN IN SILVERLIGHT 2 2

2.1. Focus 2

2.2. GetIthSubImage 2

2.3. Select 2

2.4. SelectAll 2

3. NEUE PROPERTIES IN SILVERLIGHT 2 2

3.1. AcceptsReturn 2

3.2. AspectRatio 2

3.3. BorderBrush 2

3.4. BorderThickness 2

3.5. ColumnDefinitions 2

3.6. DroppedFramesPerSecond 2

3.7. Grid.Column 2

3.8. Grid.ColumnSpan 2

3.9. Grid.Row 2

3.10. Grid.RowSpan 2

3.11. GridUnitType 2

3.12. Handled 2

3.13. Height (RowDefinition) 2

3.14. HorizontalAlignment 2

3.15. HorizontalOffset 2

3.16. IsEnabled 2

3.17. IsOpen 2

3.18. IsReadOnly 2

3.19. IsTabStop 2

3.20. KeyFrames (ObjectAnimationUsingKeyFrames) 2

3.21. Language (UIElement) 2

3.22. LineStackingStrategy 2

3.23. Margin 2

3.24. MaxHeight 2

3.25. MaxLength 2

3.26. MaxWidth 2

3.27. MinHeight 2

3.28. MinWidth 2

3.29. Opacity (MultiScaleSubImage) 2

3.30. Orientation 2

3.31. Padding 2

3.32. Password 2

3.33. PasswordChar 2

3.34. RenderedFramesPerSecond 2

3.35. SelectedText 2

3.36. SelectionBackground 2

3.37. SelectionForeground 2

3.38. SelectionLength 2

3.39. SelectionStart 2

3.40. Setters 2

3.41. ShowGridLines 2

3.42. Source (MultiScaleImage) 2

3.43. Style (UIElement) 2

3.44. TabIndex 2

3.45. TabNavigation 2

3.46. Text (TextBox) 2

3.47. TextAlignment 2

3.48. UriSource (BitmapImage) 2

3.49. UriSource (DeepZoomImageTileSource) 2

3.50. UseLayoutRounding 2

3.51. UseSprings 2

3.52. Value (DiscreteObjectKeyFrame) 2

3.53. Value (GridLength) 2

3.54. Value (Setter) 2

3.55. VerticalAlignment 2

3.56. VerticalOffset 2

3.57. ViewportOrigin 2

3.58. ViewportWidth 2

3.59. Width (ColumnDefinition) 2

3.60. ZIndex (MultiScaleSubImage) 2

4. NEUE EVENTS IN SILVERLIGHT 2 2

4.1. Closed 2

4.2. ImageOpenFailed 2

4.3. ImageOpenSucceeded 2

4.4. MotionFinished 2

4.5. Opened 2

4.6. PasswordChanged 2

4.7. SelectionChanged 2

4.8. TextChanged 2

4.9. ViewportChanged 2

5. NEUE ENUMERATIONS IN SILVERLIGHT 2 2

5.1. HorizontalAlignment Enumeration 2

5.2. KeyboardNavigationMode 2

5.3. TextAlignment Enumeration 2

5.4. VerticalAlignment Enumeration 2

6. SILVERLIGHT.JS REFERENZ 2

6.1. createObject 2

6.2. createObjectEx 2

6.3. default_error_handler 2

6.4. getSilverlight 2

6.5. isBrowserRestartRequired 2

6.6. isInstalled 2

6.7. onGetSilverlight 2

6.8. onSilverlightInstalled 2

6.9. WaitForInstallCompletion 2

7. SILVERLIGHT ERROR-MESSAGES 2

7.1. Script Errors in the DOM 2

7.2. Native Code Basis of the Silverlight Plug-in 2

7.3. Errors from Scripts, Tools, or Dialogs 2

7.4. Silverlight Plug-in Error Messages 2

7.5. Warning: Web services at '<URL>' might not be accessible due to cross-domain restrictions 2

7.6. Warning: Request for cross-domain policy at '<URL>' timed out 2

7.7. Warning: Could not locate cross-domain policy at '<URL>' 2

8. SILVERLIGHT PLUG-IN OBJECT REFERENCE 2

8.1. Property Values of the HTML Object Element 2

8.2. ActualHeight (Silverlight Plug-in Object) 2

8.3. ActualWidth (Silverlight Plug-in Object) 2

8.4. AllowHtmlPopupWindow (Silverlight Plug-in Object) 2

8.5. AutoUpgrade (Silverlight Plug-in Object) 2

8.6. Background (Silverlight Plug-in Object) 2

8.7. EnableFramerateCounter (Silverlight Plug-in Object) 2

8.8. EnableHtmlAccess (Silverlight Plug-in Object) 2

8.9. EnableRedrawRegions (Silverlight Plug-in Object) 2

8.10. FullScreen (Silverlight Plug-in Object) 2

8.11. InitParams (Silverlight Plug-in Object) 2

8.12. IsLoaded (Silverlight Plug-in Object) 2

8.13. MaxFrameRate (Silverlight Plug-in Object) 2

8.14. MinRuntimeVersion (Silverlight Plug-in Object) 2

8.15. OnError (Silverlight Plug-in Object) 2

8.16. OnFullScreenChanged (Silverlight Plug-in Object) 2

8.17. OnLoad (Silverlight Plug-in Object) 2

8.18. OnResize (Silverlight Plug-in Object) 2

8.19. OnSourceDownloadComplete (Silverlight Plug-in Object) 2

8.20. OnSourceDownloadProgressChanged (Silverlight Plug-in Object) 2

8.21. Source (Silverlight Plug-in Object) 2

8.22. SplashScreenSource (Silverlight Plug-in Object) 2

8.23. Windowless (Silverlight Plug-in Object) 2

1. neue Objekte in Silverlight 2

1.1. BitmapImage

Provides the practical object source type for the Image.Source and ImageBrush.ImageSource properties.

XAML
```
<BitmapImage .../>
```
JScript
```
To create an object using scripting, see CreateFromXAML.
```
⊟ Methods

Equals

FindName (DependencyObject)

GetHost

GetValue

SetValue

⊟ Properties

Name (DependencyObject)

UriSource

⊟ Managed Equivalent

BitmapImage

⊟ Version Information

Silverlight 2

1.2. Border

Draws a border, background, or both around another object.

XAML

```
<Border ...>
    singleChild
</Border>
-or-
<Border .../>
```

JScript

```
To create an object by using scripting, see CreateFromXAML.
```

XAML Values

Value	Description
singleChild	A single object that derives from UIElement. These can be one or more of the following: Border, Canvas, Ellipse, Glyphs, Grid, Image, InkPresenter, Line, MediaElement, PasswordBox, Path, Polygon, Polyline, Rectangle, Shape, StackPanel, TextBlock, TextBox.

Methods

AddEventListener

CaptureMouse

Equals

FindName (DependencyObject)

GetHost

GetParent

GetValue

ReleaseMouseCapture

RemoveEventListener

SetValue

⊟ Properties

ActualHeight (UIElement)

ActualWidth (UIElement)

Background (Elements)

BorderBrush

BorderThickness

Canvas.Left

Canvas.Top

Canvas.ZIndex

Child

Clip

CornerRadius (Border)

Cursor

Grid.Column

Grid.ColumnSpan

Grid.Row

Grid.RowSpan

Height (UIElement)

HorizontalAlignment

Language (UIElement)

Margin

MaxHeight

MaxWidth

MinHeight

MinWidth

Name (DependencyObject)

Opacity (UIElement)

OpacityMask

Padding

RenderTransform

RenderTransformOrigin

Resources

Style (UIElement)

Tag

Triggers

VerticalAlignment

Visibility

Width (UIElement)

⊟ Events

GotFocus

Loaded

LostFocus

MouseEnter

MouseLeave

MouseLeftButtonDown

MouseLeftButtonUp

MouseMove

⊟ Managed Equivalent

Border

⊟ Version Information

Silverlight 2

1.3. ColumnDefinition

Defines column-specific properties that apply to a Grid object.

XAML
```
<ColumnDefinition .../>
```
JScript
```
To create an object using scripting, see CreateFromXAML.
```
⊟ Methods

Equals

FindName

GetHost

GetValue

SetValue

⊟ Properties

ActualWidth

MaxWidth

MinWidth

Name

Width

⊟ Managed Equivalent

ColumnDefinition

⊟ Version Information

Silverlight 2

1.4. CornerRadius

Describes the characteristics of a rounded corner, such as can be applied to a Border.

XAML
```
<object property = "cornerRadiusString" ... />
```
JScript
```
object.property = "cornerRadiusString"
```
⊟ cornerRadiusString Grammar

uniformRadius

- or -

topLeft, *topRight*, *bottomRight*, *bottomLeft*

uniform	A value that specifies a uniform radius. The *uniformRadius* value is applied to all four corners of the CornerRadius property
topLeft, *topRight*,	Numeric values that specify the four individual radius values of a CornerRadius.

bottomRight,

bottomLeft

⊟ Managed Equivalent

CornerRadius

⊟ Version Information

Silverlight 2

1.5. DeepZoomImageTileSource

Used to specify the source of a MultiScaleImage.

XAML
```
<DeepZoomImageTileSource .../>
```
JScript
```
To create an object using scripting, see CreateFromXAML.
```
⊟ Methods

Equals, FindName (DependencyObject), GetHost, GetValue, SetValue

⊟ Properties

Name (DependencyObject), UriSource (DeepZoomImageTileSource)

⊟ Managed Equivalent

DeepZoomImageTileSource

⊟ Version Information

Silverlight 2

1.6. DiscreteObjectKeyFrame

Animates from the object value of the previous key frame to the Value of the current key frame, by using discrete values.

XAML

```
<DiscreteObjectKeyFrame .../>
```

JScript

```
To create an object using scripting, see CreateFromXAML.
```

⊟ Methods

Equals

GetValue

GetHost

FindName (DependencyObject)

SetValue

⊟ Properties

KeyTime (PointKeyFrame)

Name (DependencyObject)

Value (DiscreteObjectKeyFrame)

⊟ Managed Equivalent

DiscreteObjectKeyFrame

⊟ Version Information

Silverlight 2

1.7. Grid

Defines a grid area that consists of columns and rows.

XAML
```
<Grid ...>
    oneOrMoreUIElements
</Grid>
-or-
<Grid .../>
```
JScript
```
To create an object using scripting, see the CreateFromXAML method.
```
⊟ XAML Values

Value	Description
oneOrMoreUIElements	One or more of the following object elements that derive from UIElement: Border, Canvas, Ellipse, Glyphs, Grid, Image, InkPresenter, Line, MediaElement, PasswordBox, Path, Polygon, Polyline, Popup, Rectangle, Shape, StackPanel, TextBlock, TextBox.

⊟ Methods

AddEventListener

CaptureMouse

Equals

FindName (DependencyObject)

GetHost

GetParent

GetValue

ReleaseMouseCapture

RemoveEventListener

SetValue

⊟ Properties

Background (Elements)

Canvas.Left

Canvas.Top

Canvas.ZIndex

Children (Panel)

Clip

ColumnDefinitions

Cursor

Grid.Column

Grid.ColumnSpan

Grid.Row

Grid.RowSpan

Height (UIElement)

HorizontalAlignment

IsHitTestVisible

Language (UIElement)

Margin

MaxHeight

MaxWidth

MinHeight

MinWidth

Name (DependencyObject)

Opacity (UIElement)

OpacityMask

RenderTransform

RenderTransformOrigin

Resources

RowDefinitions

ShowGridLines

Style (UIElement)

Tag

Triggers

UseLayoutRounding

VerticalAlignment

Visibility

Width (UIElement)

⊟ Events

GotFocus

KeyDown

KeyUp

Loaded

LostFocus

MouseEnter

MouseLeave

MouseLeftButtonDown

MouseLeftButtonUp

MouseMove

⊟ Managed Equivalent

Grid

⊟ Version Information

Silverlight 2

1.8. GridLength

Represents the length of elements that support Star unit types.

XAML
```
<object property = "gridLengthString" ... />
```
JScript
```
object.property = "gridLengthString"
```
⊟ gridLengthString Grammar

doubleValue

- or -

starSizing

-or-

Auto

doubleValue A row's height or a column's width, in pixels.

starSizing The size expressed as a measure to fit rows or columns to available space. Star sizing includes the asterisk character (*), and optionally precedes the asterisk with an integer value that specifies a weighted factor versus other possible star sizings (for example, "3*").

Auto The literal "Auto". Divides space that is not already claimed by other rows or columns.

⊟ Properties

GridUnitType

Value

⊟ Managed Equivalent

GridLength

⊟ Version Information

Silverlight 2

1.9. MouseButtonEventArgs

Provides data for events that involve mouse buttons.

XAML

```
Cannot be used in XAML.
```

JScript

```
Obtained as arguments for certain event handlers.
```

⊟ Methods

Equals

GetStylusPoints

GetValue

GetHost

GetPosition

GetStylusInfo

FindName

SetValue

⊟ Properties

Ctrl

Handled

Name

Shift

Source

⊟ Managed Equivalent

MouseButtonEventArgs

⊟ Remarks

MouseEventArgs is used with the following events: MouseLeftButtonDown

MouseLeftButtonUp

1.10. MultiScaleImage

Enables users to open a multi-resolution image which can be zoomed in on and panned across.

XAML

```
<MultiScaleImage .../>
```

JScript

```
To create an object by using scripting, see CreateFromXAML.
```

⊟ Methods

AddEventListener

CaptureMouse

Equals

FindName (DependencyObject)

GetHost

GetIthSubImage

GetParent

GetValue

ReleaseMouseCapture

RemoveEventListener

SetValue

⊟ Properties

ActualHeight (UIElement)

ActualWidth (UIElement)

AspectRatio

Canvas.Left

Canvas.Top

Canvas.ZIndex

Clip

Cursor

Grid.Column

Grid.ColumnSpan

Grid.Row

Grid.RowSpan

Height (UIElement)

HorizontalAlignment

Language (UIElement)

Margin

MaxHeight

MaxWidth

MinHeight

MinWidth

Name (DependencyObject)

Opacity (UIElement)

OpacityMask

RenderTransform

RenderTransformOrigin

Resources

Source (MultiScaleImage)

Style (UIElement)

Tag

Triggers

UseSprings

VerticalAlignment

ViewportOrigin

ViewportWidth

Visibility

Width (UIElement)

⊟ Events

ImageFailed (MultiScaleImage)

ImageOpenFailed

ImageOpenSucceeded

Loaded

MotionFinished

MouseEnter

MouseLeave

MouseLeftButtonDown

MouseLeftButtonUp

MouseMove

ViewportChanged

⊟ Managed Equivalent

MultiScaleImage

⊟ Version Information

Silverlight 2

1.11. MultiScaleSubImage

This class holds the properties for each subimage within a MultiScaleImage object.

XAML
```
Cannot be used in XAML.
```
JScript
```
See GetIthSubImage.
```
⊟ Methods

Equals

FindName (DependencyObject)

GetHost

GetValue

SetValue

⊟ Properties

AspectRatio

Name (DependencyObject)

Opacity (MultiScaleSubImage)

ViewportOrigin

ViewportWidth

ZIndex (MultiScaleSubImage)

⊟ Managed Equivalent

MultiScaleSubImage

⊟ Version Information

Silverlight 2

1.13. ObjectAnimationUsingKeyFrames

Animates the value of an object-type property along a set of key-frames over a specified duration.

```
<ObjectAnimationUsingKeyFrames>
    oneOrMoreDiscreteObjectKeyFrames
</ObjectAnimationUsingKeyFrames>
```

JScript

```
To create an object using scripting, see CreateFromXAML.
```

⊟ XAML Values

Value	Description
oneOrMoreDiscreteObjectKeyFrames	One or more DiscreteObjectKeyFrame object elements that define the key frames for the animation.

⊟ Methods

Equals

FindName (DependencyObject)

GetHost

GetValue

SetValue

⊟ Properties

AutoReverse

BeginTime

Duration (Timeline)

FillBehavior

KeyFrames (ObjectAnimationUsingKeyFrames)

Name (DependencyObject)

RepeatBehavior (Timeline)

SpeedRatio

Storyboard.TargetName

Storyboard.TargetProperty

⊟ Managed Equivalent

ObjectAnimationUsingKeyFrames

⊟ Version Information

Silverlight 2

1.14. ObjectKeyFrameCollection

Represents a collection of DiscreteObjectKeyFrame objects that can be individually accessed by index.

XAML

```
<ObjectKeyFrameCollection ...>
    oneOrMoreDiscreteObjectKeyFrames
</ObjectKeyFrameCollection>
```

JScript

```
To create an object using scripting, see CreateFromXAML.
```

XAML Values

Value	Description
oneOrMoreDiscreteObjectKeyFrames	One or more DiscreteObjectKeyFrame object elements that define the key frames for the animation.

Methods

Add

Clear

Equals

FindName (DependencyObject)

GetHost

GetItem

GetValue

Insert

Remove

RemoveAt

SetValue

Properties

Count

Name (DependencyObject)

⊟ Managed Equivalent

ObjectKeyFrameCollection

⊟ Version Information

Silverlight 2

1.15. PasswordBox

Represents a control for entering passwords.

XAML
```
<PasswordBox   .../>
```
JScript
```
To create an object using scripting, see CreateFromXAML.
```
⊟ Methods

AddEventListener

CaptureMouse

Equals

FindName (DependencyObject)

Focus

GetHost

GetParent

GetValue

ReleaseMouseCapture

RemoveEventListener

SelectAll

SetFontSource

SetValue

⊟ Properties

AcceptsReturn

ActualHeight (UIElement)

ActualWidth (UIElement)

Background (Elements)

Canvas.Left

Canvas.Top

Canvas.ZIndex

Clip

Cursor

FontFamily

FontSize

FontStretch

FontStyle

FontWeight

Foreground

Grid.Column

Grid.ColumnSpan

Grid.Row

Grid.RowSpan

Height (UIElement)

HorizontalAlignment

IsEnabled

IsTabStop

Language (UIElement)

Margin

MaxHeight

MaxWidth

MinHeight

MinWidth

Name (DependencyObject)

Opacity (UIElement)

OpacityMask

Padding

Password

PasswordChar

RenderTransform

RenderTransformOrigin

Resources

SelectionBackground

SelectionForeground

Style (UIElement)

TabIndex

TabNavigation

Tag

Triggers

VerticalAlignment

Visibility

Width (UIElement)

⊟ Events

GotFocus

KeyDown

KeyUp

Loaded

LostFocus

MouseEnter

MouseLeftButtonDown

MouseLeave

MouseLeftButtonUp

MouseMove

PasswordChanged

⊟ Managed Equivalent

PasswordBox

⊟ Version Information

Silverlight 2

1.16. Popup

Displays content on top of existing Silverlight content, within the bounds of the Silverlight content area.

XAML

```
<Popup ...>
    singleChild
</Popup>
-or-
<Popup .../>
```

JScript

```
To create an object using scripting, see CreateFromXAML.
```

⊟ XAML Values

Value	Description
singleChild	A single object that derives from UIElement. This can be one of the following: Border, Canvas, Ellipse, Glyphs, Grid, Image, InkPresenter, Line, MediaElement, PasswordBox, Path, Polygon, Polyline, Rectangle, Shape, StackPanel, TextBlock, TextBox.

⊟ Methods

AddEventListener

CaptureMouse

Equals

FindName (DependencyObject)

GetHost

GetParent

GetValue

ReleaseMouseCapture

RemoveEventListener

SetValue

⊟ Properties

ActualHeight (UIElement)

ActualWidth (UIElement)

Background (Elements)

BorderBrush

BorderThickness

Canvas.Left

Canvas.Top

Canvas.ZIndex

Child (Popup)

Clip

CornerRadius (Border)

Cursor

Grid.Column

Grid.ColumnSpan

Grid.Row

Grid.RowSpan

Height (UIElement)

HorizontalAlignment

HorizontalOffset

IsOpen

Language (UIElement)

Margin

MaxHeight

MaxWidth

MinHeight

MinWidth

Name (DependencyObject)

Opacity (UIElement)

OpacityMask

Padding

RenderTransform

RenderTransformOrigin

Resources

Style (UIElement)

Tag

Triggers

VerticalAlignment

VerticalOffset

Visibility

Width (UIElement)

⊟ Events

Closed

Loaded

MouseEnter

MouseLeave

MouseLeftButtonDown

MouseLeftButtonUp

MouseMove

Opened

⊟ Managed Equivalent

Popup

1.17. RowDefinition

Defines row-specific properties that apply to the Grid class.

XAML
```
<RowDefinition .../>
```
JScript
```
To create an object using scripting, see CreateFromXAML.
```
⊟ Methods

Equals

FindName

GetHost

GetValue

SetValue

⊟ Properties

ActualHeight

Height

Name

MaxHeight

MinHeight

⊟ Managed Equivalent

RowDefinition

⊟ Version Information

Silverlight 2

1.18. Setter

Applies a value to a property in a Style.

XAML
```
<Setter .../>
```
JScript
```
To create an object using scripting, see CreateFromXAML.
```
⊟ Methods

Equals

FindName (DependencyObject)

GetHost

GetValue

SetValue

⊟ Properties

Name (DependencyObject)

Property

Value

⊟ Managed Equivalent

Setter

⊟ Remarks

Styles have significant restrictions in the JavaScript API for Silverlight.

⊟ Version Information

Silverlight 2

1.19. StackPanel

Arranges child elements into a single line that can be oriented horizontally or vertically.

XAML

```
<StackPanel ...>
    oneOrMoreUIElements
</StackPanel>
```

```
-or-
<StackPanel .../>
```

JScript

```
To create an object using scripting, see the CreateFromXAML method.
```

XAML Values

Value	Description
oneOrMoreUIElements	One or more of the following object elements that derive from UIElement: Border, Canvas, Ellipse, Glyphs, Grid, Image, InkPresenter, Line, MediaElement, PasswordBox, Path, Polygon, Polyline, Popup, Rectangle, StackPanel, TextBlock, TextBox.

Methods

AddEventListener

CaptureMouse

Equals

FindName (DependencyObject)

GetHost

GetParent

GetValue

ReleaseMouseCapture

RemoveEventListener

SetValue

Properties

Background (Elements)

Canvas.Left

Canvas.Top

Canvas.ZIndex

Children (Panel)

Clip

Cursor

Grid.Column

Grid.ColumnSpan

Grid.Row

Grid.RowSpan

Height (UIElement)

HorizontalAlignment

IsHitTestVisible

Language (UIElement)

Margin

MaxHeight

MaxWidth

MinHeight

MinWidth

Name (DependencyObject)

Opacity (UIElement)

Orientation

OpacityMask

RenderTransform

RenderTransformOrigin

Resources

Style (UIElement)

Tag

Triggers

UseLayoutRounding

VerticalAlignment

Visibility

Width (UIElement)

⊟ Events

GotFocus

KeyDown

KeyUp

Loaded

LostFocus

MouseEnter

MouseLeave

MouseLeftButtonDown

MouseLeftButtonUp

MouseMove

⊟ Managed Equivalent

StackPanel

⊟ Version Information

Silverlight 2

1.20. Style (Object)

Contains property setters that can be shared between instances of a type.

XAML
```
<Style ...>
```

```
    oneOrMoreSetters
</Style>
-or-
<Style .../>
```

JScript

```
To create an object using scripting, see CreateFromXAML.
```

XAML Values

Value	Description
oneOrMoreSetters	One or more Setter object elements.

Methods

Equals

FindName (DependencyObject)

GetHost

GetValue

SetValue

Properties

Name (DependencyObject)

Setters

Managed Equivalent

Style

Remarks

Styles have significant restrictions in the JavaScript API for Silverlight.

⊟ Version Information

Silverlight 2

1.21. TextBox

Represents a control that can be used to display editable text.

XAML
```
<TextBox .../>
```
JScript
```
To create an object using scripting, see CreateFromXAML.
```
⊟ Methods

AddEventListener

CaptureMouse

Equals

FindName (DependencyObject)

Focus

GetHost

GetParent

GetValue

ReleaseMouseCapture

RemoveEventListener

Select

SelectAll

SetFontSource

SetValue

⊟ Properties

AcceptsReturn

ActualHeight (UIElement)

ActualWidth (UIElement)

Background (Elements)

Canvas.Left

Canvas.Top

Canvas.ZIndex

Clip

Cursor

FontFamily

FontSize

FontStretch

FontStyle

FontWeight

Foreground

Grid.Column

Grid.ColumnSpan

Grid.Row

Grid.RowSpan

Height (UIElement)

HorizontalAlignment

IsEnabled

IsReadOnly

IsTabStop

Language (UIElement)

Margin

MaxHeight

MaxLength

MaxWidth

MinHeight

MinWidth

Name (DependencyObject)

Opacity (UIElement)

OpacityMask

Padding

RenderTransform

RenderTransformOrigin

Resources

SelectedText

SelectionBackground

SelectionForeground

SelectionLength

SelectionStart

Style (UIElement)

TabIndex

TabNavigation

Tag

Text (TextBox)

TextAlignment

TextDecorations

TextWrapping

Triggers

VerticalAlignment

Visibility

Width (UIElement)

⊟ Events

GotFocus

KeyDown

KeyUp

Loaded

LostFocus

MouseEnter

MouseLeave

MouseLeftButtonDown

MouseLeftButtonUp

MouseMove

SelectionChanged

TextChanged

⊟ Managed Equivalent

TextBox

⊟ Version Information

Silverlight 2

1.22. Thickness

Describes the thickness of a frame around a rectangle.

XAML
```
<object property = "thicknessString" ... />
```
JScript
```
object.property = "thicknessString"
```
⊟ thicknessString Grammar

```
        uniform
- or -
leftRight,topBottom
- or -
left,top,right,bottom
```
uniform Specifies a uniform thickness, in pixels.

lefRright	Specifies the left and right thickness, in pixels.
topBottom	Specifies the top and bottom thickness, in pixels.
left top right bottom	Specifies the thickness of the individual sides, in pixels.

⊟ Managed Equivalent

Thickness

⊟ Version Information

Silverlight 2

2. neue Methoden in Silverlight 2

2.1. Focus

Attempts to set the focus on the control.

XAML
```
Cannot use methods in XAML.
```
JScript
```
retval = object.Focus()
```
⊟ Return Value

Type: Boolean

true if focus was set to the control, or focus was already on the control. false if the control is not focusable.

⊟ Managed Equivalent

Focus

⊟ Applies To

TextBox, PasswordBox

⊟ Version Information

Silverlight 2

2.2. GetIthSubImage

Retrieves a single MultiScaleSubImage object from within the multiresolution image used by the MultiScaleImage.

XAML
```
Cannot use methods in XAML.
```

JScript
```
retval = object.GetIthSubImage(index)
```
⊟ Arguments

integer
index

The index at which the object should be retrieved.

⊟ Return Value

Type: MultiScaleSubImage

A reference to the object if successful; otherwise, null.

⊟ Managed Equivalent

SubImages

⊟ Applies To

MultiScaleImage

2.3. Select

Selects a range of text in the text box.

XAML
```
Cannot use methods in XAML.
```
JScript
```
object.Select(start, length)
```
⊟ Arguments

int
start

The zero-based index of the first character in the selection.

int

length

The length of the selection, in characters.

Managed Equivalent

Select

Applies To

TextBox

Version Information

Silverlight 2

2.4. SelectAll

Selects the entire contents of the textbox (or password box).

XAML

```
Cannot use methods in XAML.
```

JScript

```
object.SelectAll()
```

Managed Equivalent

TextBox..::.SelectAll; PasswordBox..::.SelectAll

Applies To

TextBox, PasswordBox

Version Information

Silverlight 2

3. neue Properties in Silverlight 2

3.1. AcceptsReturn

Gets or sets a value that determines whether a text box accepts newline and carriage-return characters.

XAML
```
<TextBox AcceptsReturn="bool" .../>
```
JScript
```
value = object.AcceptsReturn
object.AcceptsReturn = value
```
⊟ Property Value

Type: Boolean

true if the text box accepts newline characters; otherwise, false.

This property is read/write. The default value is false.

⊟ Managed Equivalent

AcceptsReturn

⊟ Remarks

When AcceptsReturn is true, newline and carriage-return characters result in a newline applied to the displayed text.

⊟ Applies To

TextBox

⊟ Version Information

Silverlight 2

3.2. AspectRatio

Gets the aspect ratio of the image used as the source.

XAML
```
Cannot be set in XAML.
```
JScript
```
value = object.AspectRatio
```
⊟ Property Value

Type: double

The width of the image divided by its height.

This property is read-only.

⊟ Managed Equivalent

AspectRatio

⊟ Applies To

MultiScaleImage, MultiScaleSubImage

⊟ Version Information

Silverlight 2

3.3. BorderBrush

Gets or sets the brush that fills the border area.

XAML
```
<object BorderBrush="colorString" .../>
```

```
-or-
<object>
  <object.BorderBrush>
        singleBrush
  </object.BorderBrush>
</object>
```

JScript

```
value = object.BorderBrush
object.BorderBrush = value
```

⊟ XAML Values

See Brush.

⊟ Property Value

Type: Brush

The brush that fills the border area.

This property is read/write. The default is a null brush.

⊟ Managed Equivalent

Control..::.BorderBrush; Border..::.BorderBrush

⊟ Applies To

Border, TextBox, PasswordBox

⊟ Version Information

Silverlight 2

3.4. BorderThickness

Gets or sets the dimensions of a border area.

XAML
```
<object BorderThickness="uniform"/>
- or -
<object BorderThickness="leftright,topbottom"/>
- or -
<object BorderThickness="left,top,right,bottom"/>
```
JScript
```
value = object.BorderThickness
object.BorderThickness = value
```
⊟ Property Value

Type: Thickness

An object that defines the four measurements of the border area.

This property is read/write. The default is an empty object, which evaluates to all values as 0.

⊟ XAML Values

See Thickness.

⊟ Managed Equivalent

Border..::.BorderThickness; Control..::.BorderThickness

⊟ Applies To

Border, TextBox, PasswordBox

⊟ Version Information

Silverlight 2

3.5. ColumnDefinitions

Holds column definitions for a Grid.

XAML

```
<Grid ...>
  <Grid.ColumnDefinitions>
      oneOrMoreColumnDefinitions
  </Grid.ColumnDefinitions>
</Grid>
```

⊟ XAML Values

Value	Description
oneOrMoreColumnDefinitions	One or more ColumnDefinition object elements, in the form <ColumnDefinition/>

⊟ Property Value

Type: Collection of ColumnDefinition

Not typically used in scripting.

⊟ Managed Equivalent

ColumnDefinitions

⊟ Applies To

Grid

⊟ Version Information

Silverlight 2

3.6. DroppedFramesPerSecond

Gets the number of frames per second that are being dropped by the media.

XAML
```
Cannot be used in XAML.
```
JScript
```
value = object.DroppedFramesPerSecond
```
⊟ Property Value

Type: double

The number of frames per second that are being dropped by the media.

This property is read-only.

⊟ Managed Equivalent

DroppedFramesPerSecond

⊟ Applies To

MediaElement

⊟ Version Information

Silverlight 2

3.7. Grid.Column

Gets or sets a value that indicates the column in which child content in a Grid object should appear.

XAML
```
<object Grid.Column="integer"  .../>
```

JScript
```
value = object["Grid.Column"]
object["Grid.Column"] = value
```
⊟ Property Value

Type: integer

An integer that specifies which column in a Grid instance (by zero-based index) should render the object where Grid.Column is applied as an attached property.

This property is read/write. The default is 0.

⊟ Managed Equivalent

Grid..::.Column

⊟ Remarks

Grid.Column is an attached property.

⊟ Applies To

UIElement

⊟ Version Information

Silverlight 2

3.8. Grid.ColumnSpan

Gets or sets a value that indicates the total number of columns that child content spans within a Grid object.

XAML
```
<object Grid.ColumnSpan="integer"  .../>
```

JScript
```
value = object["Grid.ColumnSpan"]
object["Grid.ColumnSpan"] = value
```
⊟ Property Value

Type: integer

A value that represents the number of columns to span, when the object is positioned where Grid.ColumnSpan is applied as an attached property.

This property is read/write. The default value is 1.

⊟ Managed Equivalent

Grid..::.ColumnSpan

⊟ Remarks

Grid.ColumnSpan is an attached property.

⊟ Applies To

UIElement

⊟ Version Information

Silverlight 2

3.9. Grid.Row

Gets or sets a value that indicates the row in which child content within a Grid object should appear.

XAML
```
<object Grid.Row="integer"  .../>
```
JScript

```
value = object["Grid.Row"]
object["Grid.Row"] = value
```

Property Value

Type: integer

An integer that specifies which row in a Grid instance (by zero-based index) should render the object where Grid.Row is applied as an attached property.

This property is read/write. The default is 0.

Managed Equivalent

Grid..::.Row

Remarks

Grid.Row is an attached property.

Applies To

UIElement

Version Information

Silverlight 2

3.10. Grid.RowSpan

Gets or sets a value that indicates the total number of rows that child content spans within a Grid.

XAML

```
<object Grid.RowSpan="integer"  .../>
```

JScript

```
value = object["Grid.RowSpan"]
object["Grid.RowSpan"] = value
```

Property Value

Type: integer

A value that represents the number of rows to span, when the object is positioned where Grid.RowSpan is applied as an attached property.

This property is read/write. The default value is 1.

Managed Equivalent

Grid..::.RowSpan

Remarks

Grid.RowSpan is an attached property.

Applies To

UIElement

Version Information

Silverlight 2

3.11. GridUnitType

Gets the associated unit type for the GridLength.

XAML

```
Cannot be used in XAML.
```

JScript

```
value = object.GridUnitType
```

⊟ Property Value

Type: Enumeration

- Auto The size is determined by the size properties of the content object.
- Pixel The value is expressed in pixels.
- Star The value is expressed as a weighted proportion of available space.

This property is read-only.

⊟ Managed Equivalent

GridUnitType

⊟ Applies To

GridLength

⊟ Version Information

Silverlight 2

3.12. Handled

Gets or sets a value that indicates whether a routed event has been handled, which can be used to prevent the event from being routed to additional objects along the event route.

XAML

```
Cannot be set in XAML.
```

JScript

```
value = eventargs.Handled
eventargs.Handled = value
```

⊟ Property Value

Type: bool

true to mark the event as handled; false to leave the event unhandled, which lets the event be routed to the next object.

This property is read/write. The default is false.

⊟ Managed Equivalent

MouseButtonEventArgs..::.Handled; KeyEventArgs..::.Handled

⊟ Applies To

KeyboardEventArgs

MouseButtonEventArgs

3.13. Height (RowDefinition)

Gets the calculated height of a RowDefinition element, or sets the GridLength value of a column that is defined by the RowDefinition element.

XAML
```
<RowDefinition Height="GridLength" .../>
```
JScript
```
value = object.Height
object.Height = value
```
⊟ XAML Values

See GridLength.

⊟ Property Value

Type: GridLength

The height, by GridLength.

This property is read/write.

⊟ Managed Equivalent

Height

⊟ Applies To

RowDefinition

⊟ Version Information

Silverlight 2

3.14. HorizontalAlignment

Gets or sets the horizontal alignment characteristics that are applied to an object when the object is composed within a layout parent, such as within a panel.

XAML
```
<object HorizontalAlignment="HorizontalAlignment" .../>
```
JScript
```
value = object.HorizontalAlignment
object.HorizontalAlignment = value
```
⊟ Property Value

Type: HorizontalAlignment

A value of the enumeration.

This property is read/write. The default is Left.

⊟ Managed Equivalent

HorizontalAlignment

⊟ Applies To

Border

Canvas

Ellipse

Glyphs

Grid

Image

InkPresenter

Line

MediaElement

PasswordBox

Path

Polygon

Polyline

Popup

Rectangle

StackPanel

TextBlock

TextBox

⊟ Version Information

Silverlight 2

3.15. HorizontalOffset

Gets or sets the distance between the left side of the Silverlight content area and the left side of the popup.

XAML
```
<Popup HorizontalOffset="double" .../>
```
JScript
```
value = object.HorizontalOffset
object.HorizontalOffset = value
```
⊟ Property Value

Type: Double

The distance, in pixels.

This property is read/write. The default is 0.

⊟ Managed Equivalent

HorizontalOffset

⊟ Applies To

Popup

⊟ Version Information

Silverlight 2

3.16. IsEnabled

Gets or sets the value that determines if the control is enabled in UI.

XAML
```
<object IsEnabled="bool" .../>
```
JScript
```
value = object.IsEnabled
object.IsEnabled = value
```
⊟ Property Value

Type: Boolean

true if the control is enabled in UI; otherwise, false.

This property is read/write. The default value is true.

⊟ Managed Equivalent

IsEnabled

⊟ Applies To

TextBox, PasswordBox

⊟ Version Information

Silverlight 2

3.17. IsOpen

Gets or sets whether the popup is currently displaying on the screen.

XAML
```
<Popup IsOpen="bool" .../>
```
JScript
```
value = object.IsOpen
object.IsOpen = value
```
⊟ Property Value

Type: Boolean

true if the popup is currently displaying; otherwise, false.

This property is read/write. The default is false.

⊟ Managed Equivalent

IsOpen

⊟ Applies To

Popup

⊟ Version Information

Silverlight 2

3.18. IsReadOnly

Gets or sets a value that determines if the text box is read-only.

XAML

```
<TextBox IsReadOnly="bool" .../>
```

JScript

```
value = object.IsReadOnly
object.IsReadOnly = value
```

Property Value

Type: Boolean

true if the text box is read-only; otherwise, false.

This property is read/write. The default value is false.

Managed Equivalent

IsReadOnly

Applies To

TextBox

Version Information

Silverlight 2

3.19. IsTabStop

Gets or sets a value that indicates whether a control is included in tab navigation.

XAML

```
<object IsTabStop="bool" .../>
```

JScript

```
value = object.IsTabStop
object.IsTabStop = value
```

Property Value

Type: Boolean

true if the control is included in tab navigation; otherwise, false.

This property is read/write. The default is true.

⊟ Managed Equivalent

IsTabStop

⊟ Applies To

TextBox

PasswordBox

⊟ Version Information

Silverlight 2

3.20. KeyFrames (ObjectAnimationUsingKeyFrames)

Gets or sets the collection of DiscreteObjectKeyFrame objects that define an animation.

XAML

```
<ObjectAnimationUsingKeyFrames...>
    oneOrMoreDiscreteObjectKeyFrames
</ObjectAnimationUsingKeyFrames>
```

JScript

```
value = object.KeyFrames
object.KeyFrames = value
```

⊟ XAML Values

Value	Description
oneOrMoreDiscreteObjectKeyFrames	One or more DiscreteObjectKeyFrame object elements that define the key frames for the animation.

⊟ Property Value

Type: ObjectKeyFrameCollection

The collection of DiscreteObjectKeyFrame objects that define the animation.

This property is read/write. The default value is an empty collection.

⊟ Managed Equivalent

KeyFrames

⊟ Applies To

PointAnimationUsingKeyFrames

⊟ Version Information

Silverlight 2

3.21. Language (UIElement)

Gets or sets the language of the content within an object.

XAML
```
<object Language="xmllangString" .../>
```
JScript
```
value = object.Language
object.Language = value
```
⊟ Property Value

Type: string

A language in the same format as xml:lang (ISO 639-1).

This property is read/write. The default is the string "en-us".

⊟ Managed Equivalent

FrameworkElement..::.Language; Inline..::.Language

⊟ Remarks

Language is typically set on root elements. It represents a meaning similar to xml:lang. Language values inherit to child elements. The practical effects of Language are limited to the following objects in Silverlight 2:

Run

TextBlock

TextBox

⊟ Applies To

Border

Canvas

Ellipse

Glyphs

Grid

Image

InkPresenter

Line

MediaElement

PasswordBox

Path

Polygon

Polyline

Popup

Rectangle

Run

StackPanel

TextBlock

TextBox

⊟ Version Information

Silverlight 2

3.22. LineStackingStrategy

Gets or sets the mechanism by which a line box is determined for each line of text in a TextBlock object.

XAML
```
<TextBlock LineStackingStrategy="MaxHeight"/>
- or -
<TextBlock LineStackingStrategy="BlockLineHeight"/>
```

JScript
```
value = object.LineStackingStrategy
object.LineStackingStrategy = value
```
⊟ Property Value

Type: Enumeration

The value is one of the following:

- MaxHeight. The stack height is based on calculating the height of the tallest line. This is the default.
- BlockLineHeight. The stack height is determined by the block-element line height.

This property is read/write.

⊟ Managed Equivalent

LineStackingStrategy

⊟ Applies To

TextBlock

⊟ Version Information

Silverlight 2

3.23. Margin

Gets or sets the outer margin of a UIElement.

XAML
```
<object Margin="uniform"/>
- or -
<object Margin="leftRight,topBottom"/>
```

```
- or -
<object Margin="left,top,right,bottom"/>
```

JScript

```
value = object.Margin
object.Margin = value
```

⊟ Property Value

Type: Thickness

An object that defines the four measurements of the margin.

This property is read/write. The default is an empty object, which evaluates to all values as 0.

⊟ XAML Values

See Thickness.

⊟ Managed Equivalent

Margin

⊟ Remarks

Margin is available only as a property for the JavaScript API if the client is Silverlight version 2.

⊟ Applies To

Border

Grid

Image

InkPresenter

Line

MediaElement

Polyline

Rectangle

StackPanel

TextBlock

TextBox

PasswordBox

Path

Ellipse

Polygon

Popup

Canvas

Glyphs

⊟ Version Information

Silverlight 2

3.24. MaxHeight

Gets or sets the maximum height constraint of an object.

XAML
```
<object MaxHeight="double" .../>
```
JScript
```
value = object.MaxHeight
object.MaxHeight = value
```
⊟ Property Value

Type: Double

The maximum height constraint.

This property is read/write. The default (for JavaScript) is an undefined large number.

⊟ Managed Equivalent

MaxHeight; MaxHeight

⊟ Applies To

Border

Canvas

Ellipse

Glyphs

Grid

Image

InkPresenter

Line

MediaElement

PasswordBox

Path

Polygon

Polyline

Popup

Rectangle

RowDefinition

StackPanel

TextBlock

TextBox

⊟ Version Information

Silverlight 2

3.25. MaxLength

Gets or sets the value that determines the maximum number of characters that are allowed for user input.

XAML
```
<object MaxLength="int" .../>
```
JScript
```
value = object.MaxLength
object.MaxLength = value
```
⊟ Property Value

Type: int

The maximum number of characters that are allowed for user input.

This property is read/write. The default is 0, which evaluates to no constraint.

⊟ Managed Equivalent

TextBox..::.MaxLength; PasswordBox..::.MaxLength

⊟ Applies To

PasswordBox

TextBox

⊟ Version Information

Silverlight 2

3.26. MaxWidth

Gets or sets the maximum width constraint of an object.

XAML
```
<object MaxWidth="double" .../>
```
JScript

```
value = object.MaxWidth
object.MaxWidth = value
```

⊟ Property Value

Type: Double

The maximum width constraint.

This property is read/write. The default (for JavaScript) is an undefined large number.

⊟ Managed Equivalent

MaxWidth; MaxWidth

⊟ Applies To

Border

Canvas

ColumnDefinition

Ellipse

Glyphs

Grid

Image

InkPresenter

Line

MediaElement

PasswordBox

Path

Polygon

Polyline

Rectangle

StackPanel

TextBlock

TextBox

⊟ Version Information

Silverlight 2

3.27. MinHeight

Gets or sets the minimum height constraint of an object.

XAML
```
<object MinHeight="double" .../>
```
JScript
```
value = object.MinHeight
object.MinHeight = value
```
⊟ Property Value

Type: Double

The minimum height constraint.

This property is read/write. The default is 0, which evaluates to no constraint.

⊟ Managed Equivalent

MinHeight; MinHeight

⊟ Applies To

Border

Canvas

Ellipse

Glyphs

Grid

Image

InkPresenter

Line

MediaElement

PasswordBox

Path

Polygon

Polyline

Popup

Rectangle

RowDefinition

StackPanel

TextBlock

TextBox

Version Information

Silverlight 2

3.28. MinWidth

Gets or sets the minimum width constraint of an object.

XAML
```
<object MinWidth="double" .../>
```
JScript
```
value = object.MinWidth
object.MinWidth = value
```
Property Value

Type: Double

The minimum width constraint.

This property is read/write. The default is 0, which evaluates to no constraint.

⊟ Managed Equivalent

MinWidth; MinWidth

⊟ Applies To

Border

Canvas

ColumnDefinition

Ellipse

Glyphs

Grid

Image

InkPresenter

Line

MediaElement

PasswordBox

Path

Polygon

Polyline

Popup

Rectangle

StackPanel

TextBlock

TextBox

Version Information

Silverlight 2

3.29. Opacity (MultiScaleSubImage)

Gets or sets the degree of the opacity of a MultiScaleSubImage object.

XAML
```
Cannot be used from XAML.
```
JScript
```
value = object.Opacity
object.Opacity = value
```
Property Value

Type: double

A value between 0 and 1.0. 1.0 indicates full opacity and 0 indicates fully transparent.

This property is read/write. The default value is 1.0.

Managed Equivalent

Opacity

⊟ Applies To

MultiScaleSubImage

⊟ Version Information

Silverlight 2

3.30. Orientation

Gets or sets the dimension by which child elements are stacked.

XAML
```
<StackPanel Orientation="Vertical" .../>
-or-
<StackPanel Orientation="Horizontal" .../>
```
JScript
```
value = object.Orientation
object.Orientation = value
```
⊟ Property Value

Type: string

A string to specify the orientation. Can be either Vertical or Horizontal.

This property is read/write. The default is Vertical.

⊟ Managed Equivalent

Orientation

⊟ Applies To

StackPanel

⊟ Version Information

Silverlight 2

3.31. Padding

Gets or sets the padding inside a container.

XAML
```
<object Padding="uniform"/>
- or -
<object Padding="leftRight,topBottom"/>
- or -
<object Padding="left,top,right,bottom"/>
```
JScript
```
value = object.Padding
object.Padding = value
```
⊟ Property Value

Type: Thickness

An object that defines the four measurements of the padding.

This property is read/write. The default is an empty object, which evaluates to all values as 0.

⊟ XAML Values

See Thickness.

⊟ Managed Equivalent

Control..::.Padding; TextBlock..::.Padding; Border..::.Padding

⊟ Remarks

Margin is only available as a property for the JavaScript API if the client is Silverlight version 2.

⊟ Applies To

Border

PasswordBox

TextBlock

TextBox

⊟ Version Information

Silverlight 2

3.32. Password

Gets or sets the password currently held by the PasswordBox.

XAML
```
<object Password="string" .../>
```
JScript
```
value = object.Password
object.Password = value
```
⊟ Property Value

Type: string

A string representing the password.

This property is read/write. The default is empty string.

⊟ Managed Equivalent

Password

⊟ Applies To

PasswordBox

⊟ Version Information

Silverlight 2

3.33. PasswordChar

Gets or sets the masking character for the PasswordBox.

XAML
```
<object PasswordChar="char" .../>
```
JScript
```
value = object.PasswordChar
object.PasswordChar = value
```
⊟ Property Value

Type: char

A single character representing the masking character.

This property is read/write. The default is empty string.

⊟ Managed Equivalent

PasswordChar

⊟ Applies To

PasswordBox

⊟ Version Information

Silverlight 2

3.34. RenderedFramesPerSecond

Gets the number of frames per second that are being rendered by the media.

XAML
```
Cannot be used in XAML.
```
JScript
```
value = object.RenderedFramesPerSecond
```
⊟ Property Value

Type: double

The number of frames per second that are being rendered by the media.

This property is read-only.

⊟ Managed Equivalent

RenderedFramesPerSecond

⊟ Applies To

MediaElement

⊟ Version Information

Silverlight 2

3.35. SelectedText

Gets or sets the content of the current selection in the text box.
XAML

<object SelectedText="string" .../>

JScript

value = object.SelectedText
object.SelectedText = value

Property Value

Type: string

The currently selected text in the text box. If no text is selected, the value is an empty string.

This property is read/write. The default is empty string.
Managed Equivalent

SelectedText
Remarks

The SelectedText property can be used to replace the selected text or to insert new text at the caret location if no text is selected. Setting this property raises the SelectionChanged event.
Applies To

TextBox
Version Information

Silverlight 2

3.36. SelectionBackground

Gets or sets the brush that fills the background of the selected text.
XAML

<object SelectionBackground="colorString" .../>
-or-
<object>

```
  <object.SelectionBackground>
       singleBrush
  </object.SelectionBackground>
</object>
```

JScript

```
value = object.SelectionBackground
object.SelectionBackground = value
```

XAML Values

See Brush.
Property Value

Type: Brush

The brush that fills the background of the selected text.

This property is read/write. The default is a null brush.
Managed Equivalent

TextBox..::.SelectionBackground; PasswordBox..::.SelectionBackground
Applies To

TextBox, PasswordBox
Version Information

Silverlight 2

3.37. SelectionForeground

Gets or sets the brush that fills the foreground of the selected text.
XAML

```
<object SelectionForeground="colorString" .../>
-or-
<object>
  <object.SelectionForeground>
```

```
      singleBrush
  </object.SelectionForeground>
</object>
```

JScript

```
value = object.SelectionForeground
object.SelectionForeground = value
```

XAML Values

See Brush.

Property Value

Type: Brush

The brush that fills the foreground of the selected text.

This property is read/write. The default is a null brush.

Managed Equivalent

TextBox..::.SelectionForeground; PasswordBox..::.SelectionForeground

Applies To

PasswordBox

TextBox

Version Information

Silverlight 2

3.38. SelectionLength

Gets or sets the number of characters in the current selection in the text box.

XAML

```
<TextBox SelectionLength="int" .../>
```

JScript

```
value = object.SelectionLength
object.SelectionLength = value
```

Property Value

Type: int

The number of characters in the current selection, or 0 if there is no selection.

This property is read/write. The default is 0.

Managed Equivalent

SelectionLength

Applies To

TextBox

Version Information

Silverlight 2

3.39. SelectionStart

Gets or sets the starting position of the text that is selected in the text box.

XAML
```
<TextBox SelectionStart="int" .../>
```
JScript
```
value = object.SelectionStart
object.SelectionStart = value
```

Property Value

Type: int

The starting position of the current selection, or 0 if there is no selection.

This property is read/write. The default is 0.

⊟ Managed Equivalent

SelectionStart

⊟ Applies To

TextBox

⊟ Version Information

Silverlight 2

3.40. Setters

Gets a collection of Setter objects.

XAML
```
<Style ...>
    oneOrMoreSetters
</Style>
```
JScript
```
value = object.Setters
```
⊟ Property Value

Type: Collection

A collection of Setter objects.

This property is read-only when you access it from script, but can be set by using XAML. The default is an empty object.

⊟ XAML Values

Value	Description
oneOrMoreSetters	One or more Setter object elements.

Setters is the implicit collection content of a Style.

⊟ Managed Equivalent

Setters

⊟ Applies To

Style

⊟ Version Information

Silverlight 2

3.41. ShowGridLines

Gets or sets a value that indicates whether grid lines are visible within the current Grid object.

XAML
```
<Grid ShowGridLines="bool" .../>
```
JScript
```
value = object.ShowGridLines
object.ShowGridLines = value
```
⊟ Property Value

Type: Boolean

true if grid lines are visible; otherwise, false.

This property is read/write. The default is false.

⊟ Managed Equivalent

ShowGridLines

⊟ Remarks

Setting ShowGridLines to true is not intended for use in production-quality code.

⊟ Applies To

Grid

⊟ Version Information

Silverlight 2

3.42. Source (MultiScaleImage)

Gets or sets the URI of the image to use for a MultiScaleImage object.

XAML
```
<MultiScaleImage Source="uri" />
-or-
<MultiScaleImage>
    <MultiScaleImage.Source>
    <DeepZoomImageTileSource UriSource="uri"/>
  </MultiScaleImage.Source>
</MultiScaleImage>
```
JScript
```
value = object.Source
object.Source = value
```
⊟ Property Value

Type: DeepZoomImageTileSource or string in XAML.

The URI.

This property is read/write.

⊟ Managed Equivalent

Source

⊟ Remarks

Specify as a string for compact usage. Specifying a DeepZoomImageTileSource is a valid verbose usage. Script access can accept or return a DeepZoomImageTileSource object.

⊟ Applies To

MultiScaleImage

⊟ Version Information

Silverlight 2

3.43. Style (UIElement)

Gets or sets an instance of the Style class that is applied for the current object during rendering.

XAML

```
<object ...>
  <object.Style>
        singleStyle
  </object.Style>
</object>
```

JScript

```
value = object.Style
object.Style = value
```

XAML Values

Value	Description
singleStyle	A single Style instance that is defined inline.

Property Value

Type: Style

A Style object.

This property is read/write.

Managed Equivalent

Style

Remarks

This property is not typically used in scripting. Styles have significant restrictions in the JavaScript API for Silverlight.

Version Information

Silverlight 2

3.44. TabIndex

Gets or sets a value that determines the order in which elements receive focus when the user navigates through controls by using the TAB key.

XAML
```
<object TabIndex="int" .../>
```
JScript
```
value = object.TabIndex
```

```
object.TabIndex = value
```

⊟ Property Value

Type: int

A value that determines the order of logical navigation for a device.

This property is read/write. The default is an indeterminate large number.

⊟ Remarks

Lower numbers are earlier in the tab sequence.

⊟ Managed Equivalent

TabIndex

⊟ Applies To

PasswordBox

TextBox

⊟ Version Information

Silverlight 2

3.45. TabNavigation

Gets or sets a value that modifies how tabbing and TabIndex work for this control.

XAML
```
<object TabNavigation="KeyboardNavigationMode" .../>
```
JScript

```
value = object.TabNavigation
object.Navigation = value
```

Property Value

Type: KeyboardNavigationMode

A value of the enumeration.

This property is read/write. The default is Local.

Managed Equivalent

TabNavigation

Applies To

TextBox, PasswordBox

Version Information

Silverlight 2

3.46. Text (TextBox)

Gets or sets the contents of the text box.

XAML
```
<TextBox Text="string" .../>
```
JScript
```
value = object.Text
object.Text = value
```

Property Value

Type: string

A string that contains the contents of the text box.

This property is read/write. The default is an empty string

⊟ Managed Equivalent

Text

⊟ Applies To

TextBox

⊟ Version Information

Silverlight 2

3.47. TextAlignment

Gets or sets how the text should be aligned.

XAML
```
<object TextAlignment="TextAlignment" .../>
```
JScript
```
value = object.TextAlignment
object.TextAlignment = value
```
⊟ Property Value

Type: TextAlignment Enumeration

A value of the enumeration.

This property is read/write. The default is Left.

⊟ Managed Equivalent

TextBox..::.TextAlignment; TextBlock..::.TextAlignment

⊟ Applies To

TextBox, TextBlock

⊟ Version Information

Silverlight 2

3.48. UriSource (BitmapImage)

Gets or sets the Uniform Resource Identifier (URI) for the source image file location.

XAML
```
<BitmapImage UriSource="string"  .../>
```
JScript
```
value = object.UriSource
object.UriSource = value
```
⊟ Property Value

Type: string

The source of the drawn image as a URI.

This property is read/write. The default value is null.

⊟ Managed Equivalent

UriSource

⊟ Remarks

The URISource property can be used to reference images in the JPEG and PNG file formats.

⊟ Applies To

BitmapImage

⊟ Version Information

Silverlight 2

3.49. UriSource (DeepZoomImageTileSource)

Gets or sets the source URI of the DeepZoomImageTileSource.

XAML
```
<DeepZoomImageTileSource UriSource="uri" />
```
JScript
```
value = object.UriSource
object.UriSource = value
```
⊟ Property Value

Type: string

A URI string that references the image metadata file.

This property is read/write.

⊟ Managed Equivalent

UriSource

⊟ Applies To

DeepZoomImageTileSource

⊟ Version Information

Silverlight 2

3.50. UseLayoutRounding

Gets or sets a value that determines whether to use rounding behavior that aligns layout to whole pixels.

XAML
```
<object UseLayoutRounding="bool" .../>
```
JScript
```
value = object.UseLayoutRounding
object.UseLayoutRounding = value
```
⊟ Property Value

Type: Boolean

true if rendering and layout should use rounding to whole pixels; otherwise, false.

This property is read/write. The default is true.

⊟ Managed Equivalent

UseLayoutRounding

⊟ Remarks

UseLayoutRounding is available for any UIElement object. However, you typically set it only on the root element in XAML.

Important Note:

If XAML and script that were originally written for Silverlight 1.0 are run on a Silverlight 2 client, UseLayoutRounding might affect rendering of the XAML. For more information, see the UseLayoutRounding property.

⊟ Applies To

Border

Canvas

Ellipse

Glyphs

Grid

Image

InkPresenter

Line

MediaElement

PasswordBox

Path

Polygon

Polyline

Rectangle

StackPanel

TextBlock

TextBox

⊟ Version Information

Silverlight 2

3.51. UseSprings

Gets or sets whether the MultiScaleImage uses spring animations.

XAML
```
<MultiScaleImage UseSprings="bool" />
```
JScript
```
value = object.UseSprings
object.UseSprings = value
```
⊟ Property Value

Type: Boolean

Whether the MultiScaleImage uses spring animations.

This property is read/write. The default value is true.

⊟ Managed Equivalent

UseSprings

⊟ Applies To

MultiScaleImage

⊟ Version Information

Silverlight 2

3.52. Value (DiscreteObjectKeyFrame)

Gets or sets the key frame's target value.

XAML

```
<DiscreteObjectKeyFrame ...>
    <DiscreteObjectKeyFrame.Value>
        singleObject
  </DiscreteObjectKeyFrame.Value>
</DiscreteObjectKeyFrame>
```

JScript

```
value = object.Value
object.Value = value
```

⊟ XAML Values

Value	Description
singleObject	A single object element of an object type that matches the animation target property type.

⊟ Property Value

Type: object

The key frame's target value, which is the value of this key frame at its specified KeyTime.

This property is read/write. The default value is null.

⊟ Managed Equivalent

Value

⊟ Applies To

DiscreteObjectKeyFrame

⊟ Version Information

Silverlight 2

3.53. Value (GridLength)

Gets a number that represents the value of the GridLength.

XAML
```
Cannot be used in XAML.
```
JScript
```
value = object.Value
```
⊟ Property Value

Type: double

The value of the GridLength.

This property is read-only.

⊟ Managed Equivalent

Value

⊟ Applies To

GridLength

⊟ Version Information

Silverlight 2

3.54. Value (Setter)

Gets or sets the value to set the styled property with.

XAML
```
<Setter ...>
```

```
  <Setter.Value>
        objectValue
  </Setter.Value>
</Setter>
-or-
<Setter Value="attributeValue"/>
```

⊟ XAML Values

objectValue

An object element that specifies the value of the property being set.

attributeValue

A string that specifies the attribute value of the property being set. The format and interpretation of the string provided for the attribute value depends on the property that is being set.

⊟ Property Value

The value to set the styled property with. For more information, see Remarks.

⊟ Managed Equivalent

Value

⊟ Remarks

This property is not typically used in scripting. Styles have significant restrictions in the JavaScript API for Silverlight.

⊟ Applies To

Setter

⊟ Version Information

Silverlight 2

3.55. VerticalAlignment

Gets or sets the vertical alignment characteristics that are applied to an object when it is composed within a layout parent, such as within a panel.

XAML
```
<object VerticalAlignment="VerticalAlignment" .../>
```
JScript
```
value = object.VerticalAlignment
object.VerticalAlignment = value
```
⊟ Property Value

Type: VerticalAlignment

A value of the enumeration.

This property is read/write. The default is Left.

⊟ Managed Equivalent

VerticalAlignment

⊟ Applies To

Border

Canvas

Ellipse

Glyphs

Grid

Image

InkPresenter

Line

MediaElement

PasswordBox

Path

Polygon

Polyline

Popup

Rectangle

StackPanel

TextBlock

TextBox

⊟ Version Information

Silverlight 2

3.56. VerticalOffset

Gets or sets the distance between the top side of the Silverlight content area and the left side of the popup.

XAML
```
<Popup VerticalOffset="double" .../>
```
JScript
```
value = object.VerticalOffset
object.VerticalOffset = value
```
⊟ Property Value

Type: Double

The distance, in pixels.

This property is read/write. The default is 0.

⊟ Managed Equivalent

VerticalOffset

⊟ Applies To

Popup

⊟ Version Information

Silverlight 2

3.57. ViewportOrigin

Gets or sets the top-left corner of the area of the image to be displayed.

XAML
```
<MultiScaleImage ViewportOrigin="Point" />
```
JScript
```
value = object.ViewportOrigin
object.ViewportOrigin = value
```

Property Value

Type: Point

A logical point that positions the viewport relative to the upper-left corner of the displayed image.

This property is read/write. The default value is a Point with the value 0,0.

Managed Equivalent

ViewportOrigin; ViewportOrigin

Applies To

MultiScaleImage

MultiScaleSubImage (scripting only)

Version Information

Silverlight 2

3.58. ViewportWidth

Gets or sets the width (zoom factor) of the area of the displayed image.

XAML
```
<MultiScaleImage ViewportWidth="double" />
```
JScript
```
value = object.ViewportWidth
object.ViewportWidth = value
```
Property Value

Type: double

The width of the area of the displayed image. This value is a zoom factor. For example, a value of 1 displays the entire image (no zoom), 0.5 is 200% zoomed in, 2 is 50% zoom, and so on.

This property is read/write.

⊟ Managed Equivalent

ViewportWidth; ViewportWidth

⊟ Applies To

MultiScaleImage

MultiScaleSubImage (scripting only)

⊟ Version Information

Silverlight 2

3.59. Width (ColumnDefinition)

Gets the calculated width of a ColumnDefinition element, or sets the GridLength value of a column that is defined by the ColumnDefinition element.

XAML
```
<object Width="GridLength" .../>
```
JScript
```
value = object.Width
object.Width = value
```
⊟ XAML Values

See GridLength.

⊟ Property Value

Type: GridLength

The width, by GridLength.

This property is read/write.

⊟ Managed Equivalent

Width

⊟ Applies To

ColumnDefinition

⊟ Version Information

Silverlight 2

3.60. ZIndex (MultiScaleSubImage)

Gets or sets a value that represents the z-order rendering behavior of the MultiScaleSubImage instance.

XAML
```
Cannot be used from XAML.
```
JScript
```
value = object.ZIndex
object.ZIndex = value
```
⊟ Property Value

Type: int

The value that represents the z-order rendering behavior.

This property is read/write. The default is 0.

⊟ Managed Equivalent

ZIndex

⊟ Remarks

Z-order determines the relative rendering order of objects (which object is on top of which other objects).

⊟ Applies To

MultiScaleSubImage

⊟ Version Information

Silverlight 2

4. neue Events in Silverlight 2

4.1. Closed

Occurs when the IsOpen property is set to false.

XAML
```
<Popup Closed="eventhandlerFunction"  .../>
```
JScript
```
[token = ]object.AddEventListener("Closed", eventhandlerFunction)
```
Arguments

Event Handler Parameters

object
sender
The object that invoked the event.

Managed Equivalent

Closed

Applies To

Popup

Version Information

Silverlight 2

4.2. ImageOpenFailed

Occurs if the first piece of metadata that is used to open the image fails. If this event occurs, no parts of the image will open successfully.

XAML

```
<MultiScaleImage ImageOpenFailed="eventhandlerFunction"  .../>
```

JScript

```
[token = ]object.AddEventListener("ImageOpenFailed", eventhandlerFunction)
```

⊟ Arguments

Event Handler Parameters

object
sender
The object that invoked the event.

⊟ Managed Equivalent

ImageOpenFailed

⊟ Applies To

MultiScaleImage

⊟ Version Information

Silverlight 2

4.3. ImageOpenSucceeded

Occurs when the first piece of metadata that is needed to load the rest of the tiles opens.

XAML

```
<MultiScaleImage ImageOpenSucceeded="eventhandlerFunction"  .../>
```

JScript

```
[token = ]object.AddEventListener("ImageOpenSucceeded", eventhandlerFunction)
```

⊟ Arguments

Event Handler Parameters

object
sender
The object that invoked the event.

Managed Equivalent

ImageOpenSucceeded

Applies To

MultiScaleImage

Version Information

Silverlight 2

4.4. MotionFinished

Occurs when the zoom or pan animation ends.

XAML
```
<MultiScaleImage MotionFinished="eventhandlerFunction"  .../>
```
JScript
```
[token = ]object.AddEventListener("MotionFinished", eventhandlerFunction)
```
Arguments

Event Handler Parameters

object
sender
The object that invoked the event.

Managed Equivalent

MotionFinished

Applies To

MultiScaleImage

Version Information

Silverlight 2

4.5. Opened

Occurs when the IsOpen property is set to true.

XAML
```
<Popup Opened="eventhandlerFunction"  .../>
```
JScript
```
[token = ]object.AddEventListener("Opened", eventhandlerFunction)
```
Arguments

Event Handler Parameters

object
sender
The object that invoked the event.

Managed Equivalent

Opened

Applies To

Popup

Version Information

Silverlight 2

4.6. PasswordChanged

Occurs when the value of the Password property changes.

XAML
```
<object PasswordChanged="eventhandlerFunction"  .../>
```
JScript
```
[token = ]object.AddEventListener("PasswordChanged", eventhandlerFunction)
```
Arguments

Event Handler Parameters

object
sender
The object that invoked the event.

Managed Equivalent

PasswordChanged

Applies To

PasswordBox

Version Information

Silverlight 2

4.7. SelectionChanged

Occurs when the text selection has changed.

XAML
```
<TextBox SelectionChanged="eventhandlerFunction"  .../>
```
JScript

```
[token = ]object.AddEventListener("SelectionChanged", eventhandlerFunction)
```

Arguments

Event Handler Parameters

object
sender
The object that invoked the event.

Managed Equivalent

SelectionChanged

Applies To

TextBox

Version Information

Silverlight 2

4.8. TextChanged

Occurs when the text selection has changed.

XAML
```
<TextBox TextChanged="eventhandlerFunction"  .../>
```
JScript
```
[token = ]object.AddEventListener("TextChanged", eventhandlerFunction)
```

Arguments

Event Handler Parameters

object
sender
The object that invoked the event.

Managed Equivalent

TextChanged

Applies To

TextBox

Version Information

Silverlight 2

4.9. ViewportChanged

Occurs when the viewport (the area of the image displayed) changes.

XAML
```
<MultiScaleImage ViewportChanged="eventhandlerFunction"  .../>
```
JScript
```
[token = ]object.AddEventListener("ViewportChanged", eventhandlerFunction)
```
Arguments

Event Handler Parameters

object
sender
The object that invoked the event.

Managed Equivalent

ViewportChanged

Applies To

MultiScaleImage

Version Information

Silverlight 2

5. neue Enumerations in Silverlight 2

5.1. HorizontalAlignment Enumeration

Indicates where an element should be displayed on the horizontal axis relative to the allocated layout slot of the parent element.

XAML
```
<object property="EnumerationValue"  .../>
```
JScript
```
value = "EnumerationValue"
```
⊟ Enumeration Values

Value	Description
Left	Aligned to the left of the layout slot.
Center	Aligned to the center of the layout slot.
Right	Aligned to the right of the layout slot.
Stretch	Stretched to fill the entire layout slot.

⊟ Managed Equivalent

HorizontalAlignment

⊟ Applies To

HorizontalAligment

⊟ Version Information

Silverlight 2

5.2. KeyboardNavigationMode

Specifies the tabbing behavior across tab stops for a Silverlight tabbing sequence within a container.

XAML
```
<object property="EnumerationValue"  .../>
```
JScript
```
value = "EnumerationValue"
```
⊟ Enumeration Values

Value	Description
Local	Tab indexes are considered on the local subtree only inside the current container.
Cycle	When the first or last keyboard navigation stop is reached, focus returns to the first or the last keyboard navigation stop inside a container.
Once	The container and its child elements as a whole receive focus only once.

⊟ Managed Equivalent

KeyboardNavigationMode

⊟ Applies To

TabNavigation

5.3. TextAlignment Enumeration

Specifies the alignment for text within a container.

XAML
```
<object property="EnumerationValue"  .../>
```
JScript
```
value = "EnumerationValue"
```
⊟ Enumeration Values

Value	Description
Center	Text is centered.
Left	Text is aligned to the left.
Right	Text is aligned to the right.

⊟ Managed Equivalent

TextAlignment

⊟ Applies To

TextAlignment

⊟ Version Information

Silverlight 2

5.4. VerticalAlignment Enumeration

Indicates where an element should be displayed on the vertical axis relative to the allocated layout slot of the parent element.

XAML
```
<object property="EnumerationValue"  .../>
```
JScript
```
value = "EnumerationValue"
```
⊟ Enumeration Values

Value	Description
Top	Aligned to the top of the layout slot.
Center	Aligned to the center of the layout slot.
Bottom	Aligned to the bottom of the layout slot.
Stretch	Stretched to fill the entire layout slot.

⊟ Managed Equivalent

VerticalAlignment

⊟ Applies To

VerticalAlignment

⊟ Version Information

Silverlight 2

6. Silverlight.js Referenz

Silverlight.js Reference

The Silverlight.js file provides JavaScript helper functions for embedding the Silverlight plug-in in a Web page and for customizing the Silverlight installation experience.

You can use the createObject and createObjectEx functions to dynamically embed a Silverlight plug-in in a Web page. These functions generate HTML object elements from the specified parameters. For more information, see How to: Add Silverlight to a Web Page using JavaScript.

You can use most of the remaining functions to improve the standard Silverlight installation or upgrade experience. For example, in some cases, you can automatically refresh the browser after the plug-in is installed, or avoid a refresh altogether. For more information, see WaitForInstallCompletion.

The Silverlight.js file is installed with the Silverlight SDK in the following location: %ProgramFiles%\Microsoft SDKs\Silverlight\v2.0\Tools. However, Silverlight.js depends on browser implementation details that can change between Silverlight releases. For this reason, it is recommended that you periodically check for an updated version from the Silverlight.js page of the MSDN Code Gallery. For more information about the Silverlight SDK, see Silverlight Tools.

Silverlight.supportedUserAgent.js is another JavaScript helper file that you can use with Silverlight.js. The Silverlight.supportedUserAgent.js file provides a supportedUserAgent function that determines whether the user's browser is supported by Silverlight. This functionality is not included in the Silverlight.js file because it is highly dependant on current Web browser implementations. Because browsers evolve very quickly, this function will need to be updated much more frequently then the Silverlight.js file. If you need the supportedUserAgent function, you should check for updates on the Silverlight.supportedUserAgent.js page regularly.

The following sections provide a brief, alphabetical reference for the Silverlight.js functions intended for developer use. This reference does not include support functions and internal helpers, which you can learn about in the Silverlight.js file itself.

The Silverlight.js file encapsulates its functions and variables within a Silverlight object, which you can retrieve from the HTML DOM window object. The following sections use the "Silverlight" prefix in the syntax blocks for this reason.

This topic contains the following sections.

- createObject
- createObjectEx

- default_error_handler
- getSilverlight
- isBrowserRestartRequired
- isInstalled
- onGetSilverlight
- onSilverlightInstalled
- WaitForInstallCompletion
- Related Topics

6.1. createObject

Generates either an HTML object element that embeds the Silverlight plug-in using the specified parameter values or alternative HTML if Silverlight is not installed.

Syntax

Silverlight.createObject(source, parentElement, id, properties, events, initParams, userContext);

-or-

objectElement = Silverlight.createObject(source, null, id, properties, events, initParams, userContext);

Parameters

source

The URI of the application to load. See Source (Silverlight Plug-in Object).

parentElement

The HTML element in which to insert the generated HTML, or null, to return the generated HTML instead of injecting it.

id

The id attribute value of the generated object element.

properties

An array of name and value pairs. See Remarks.

events

An array of event name and function name pairs, which will be added to the generated object element as child param elements.

initParams

A string that contains name and value pairs separated by commas; for example, "name1=value1,name2=value2,name3=value3".

userContext

A value that is passed to your onLoad event handler when the event occurs.

Return Value

If parentElement is null, the generated HTML object element or alternative HTML; otherwise, no value is returned.

Remarks

All parameters are optional except the source parameter.

If you specify a parentElement value, the createObject function sets the element's innerHTML property to the generated or alternative HTML. Otherwise, the createObject function returns the HTML.

The properties and events array parameters are added to the generated object element as child param elements. You can use the properties parameter to set properties including background, enableHtmlAccess, maxFrameRate, and windowless. You can use the events parameter to specify handlers for events such as onLoad and onError.

For information about these properties and events, see Silverlight Plug-in Object Reference.

You can specify a required Silverlight version by adding a properties parameter named version. For more information, see Silverlight Versioning.

To specify alternative HTML to use when Silverlight is not installed, add a properties parameter named alt with the alternative HTML as its string value. If you do not provide an alt value, the createObject function will generate default installation HTML. The default installation HTML calls the getSilverlight function and passes it the version value if you have specified one.

The createObject function provides special handling for the onLoad event. If you provide an onLoad handler, createObject will replace it with a function that calls your handler and passes it the following arguments:

- A direct reference to the generated object element.
- The userContext value.
- The onLoad event's sender argument value.

This special handling is useful to identify the loaded plug-in when you use a single onLoad handler with multiple plug-in instances in a Web page.

For more information, see How to: Add Silverlight to a Web Page using JavaScript.

6.2. createObjectEx

Calls the createObject function using parameter values specified in a single array parameter.

Syntax

Silverlight.createObjectEx(params);

-or-

objectElement = Silverlight.createObjectEx(params);

Parameters

params

The createObject parameter values as a single array. For more information, see the createObject parameter list.

Return Value

See the createObject return value.

Remarks

See the createObject remarks.

6.3. default_error_handler

Provides a default handler for the onError event that displays error data in a popup dialog box.

Syntax

Silverlight.default_error_handler = function(sender, args) { ... }

Parameters

sender

The plug-in for which the error occurred.

args

The event data. For more information, see ErrorEventArgs.

Return Value

The function that handles the event.

Remarks

You will typically suppress error handling in production code by setting default_error_handler to null.

6.4. getSilverlight

Attempts to download the specified version of the Silverlight installer.

Syntax

Silverlight.getSilverlight(version);

Parameters

version

A string that represents the requested version of Silverlight, or an empty string ("") to request the latest version. For information about the version number, see Silverlight Versioning.

Return Value

None.

Remarks

When you use createObject or createObjectEx, the default installation HTML specifies that the getSilverlight function is called when the user clicks the installation image. If you provide your own installation HTML, you should also call this function when the user requests installation.

The getSilverlight function calls the onGetSilverlight function if you have specified one. This enables you to respond when a user requests the Silverlight installer but before the installer download begins.

The getSilverlight function will then generate and follow the Silverlight installation URI for the specified Silverlight version. This will typically display a download prompt to the user without requiring navigation to a different Web page. However, if a user is on an unsupported platform, this function will navigate the browser to a Microsoft Web page that describes the problem.

6.5. isBrowserRestartRequired

Indicates whether the Silverlight plug-in is already installed when the Web pages loads, requiring a browser restart if the plug-in is upgraded.

Syntax

bool = Silverlight.isBrowserRestartRequired;

Return Value

true if the Silverlight plug-in is already installed, requiring a browser restart upon upgrade; otherwise, false.

Remarks

The WaitForInstallCompletion function calls this function to determine whether to automatically refresh the browser after Silverlight is installed.

You can use isBrowserRestartRequired to determine whether Silverlight is installed, and to display a message that a browser restart is required when you upgrade.

6.6. isInstalled

Indicates whether the specified version of Silverlight is installed.

Syntax

bool = Silverlight.isInstalled(version);

Parameters

version

A string that represents a Silverlight version number. For information about the version number, see Silverlight Versioning.

Return Value

true if the specified version of Silverlight is installed; otherwise, false.

Remarks

The createObject and createObjectEx embedding functions call the isInstalled function to determine whether a requested version of Silverlight has been installed. The return value determines whether the embedding functions will generate an HTML object element or alternative HTML that prompts the user to install or upgrade.

The WaitForInstallCompletion function calls the isInstalled function to determine whether Silverlight has been installed since the last time it checked.

6.7. onGetSilverlight

Called by the getSilverlight function to indicate that a user has requested the Silverlight installer.

Syntax

Silverlight.onGetSilverlight = function() { ... };

Parameters

None.

Return Value

The function that the getSilverlight function calls.

Remarks

When you use createObject or createObjectEx, the default installation HTML specifies that the getSilverlight function is called when the user clicks the installation image. You can also use getSilverlight with your own installation HTML.

The getSilverlight function calls the onGetSilverlight function if you have specified one. This enables you to respond to installation requests without modifying the installation HTML. The onGetSilverlight function is called before the installer download begins.

Some browsers block unintended file downloads by mapping download requests to user actions. If the onGetSilverlight function takes too long, then the browser may block the download. Therefore, it is very important for an onGetSilverlight function to avoid performing long-running operations unless you start them asynchronously. You can start asynchronous operations by using the setTimeout method.

6.8. onSilverlightInstalled

Called by the WaitForInstallCompletion function to refresh the browser when it detects that Silverlight has been installed.

Syntax

Silverlight.onSilverlightInstalled = function() { ... };

Parameters

None.

Return Value

The function that the WaitForInstallCompletion function calls.

Remarks

The WaitForInstallCompletion function calls the onSilverlightInstalled function when it detects that Silverlight has been installed. However, this does not occur when the user has upgraded Silverlight from an earlier version. In this case, the user must restart the browser.

You will typically use the isBrowserRestartRequired function to determine whether an upgrade is required, and then display the appropriate message.

The default onSilverlightInstalled function refreshes the browser window. You can replace this behavior in a handler for the Web page's onLoad event. You can disable the automatic refresh by setting the function to null, or you can set it to your own function. For example, instead of refreshing the browser, you can use createObject or createObjectEx to insert a new HTML object element directly into your Web page. This will embed the newly-installed plug-in without requiring a browser refresh.

6.9. WaitForInstallCompletion

Periodically checks for new Silverlight installations and automatically refreshes the browser window after install.

Syntax

Silverlight.WaitForInstallCompletion();

Parameters

None.

Return Value

None.

Remarks

New Silverlight installations require a browser refresh so that the Web page can load the Silverlight plug-in. However, Silverlight upgrades require a browser restart instead of just a refresh.

When you use Silverlight.js, the WaitForInstallCompletion function is called by default when the page loads and a user has not already installed Silverlight. After a new Silverlight installation is complete, the WaitForInstallCompletion function calls the onSilverlightInstalled function to refresh the browser. You can replace onSilverlightInstalled with your own function to disable or customize this behavior.

7. Silverlight Error-Messages

Silverlight
Error Messages

The primary programming model for Silverlight is a managed API, but errors (as opposed to exceptions) still exist at several points in the overall architecture and in the supporting development environment.

7.1. Script Errors in the DOM

Silverlight is ultimately a plug-in hosted in one of two possible control models (ActiveX or NPAPI). You introduce Silverlight into the host using an object element, or some technique that introduces the plug-in instance into the DOM directly.

During the phase where Silverlight is being introduced to the host, it is possible that errors are reported to scripting even before the plug-in instance can be instantiated. This might happen if you specify param elements with invalid data, or specify invalid values for an attribute of the object element such as data.

The next phase of possible errors is if values for SplashScreenSource or Source are invalid or the URIs specified cannot be resolved. Problems during this phase have not yet enabled the Silverlight infrastructure to initiate with either possible API (JavaScript or managed) and can only be reported to the DOM. The Silverlight documentation does not include reference for general DOM error messages, but troubleshooting information for initialization is included in various topics that are linked from Adding Silverlight to a Web Page using HTML or JavaScript.

If there are no errors to this point, and the managed API is now initiated by a successful package loading of Source, then most errors that originate with Silverlight can now be handled as managed application exceptions. For details, see Application Services.

7.2. Native Code Basis of the Silverlight Plug-in

If you are using the managed API, most handling has transitioned to exception handling by the application. However, there is still the possibility that certain errors that originate from the Silverlight native code will be surfaced, without first raising UnhandledException. If you are deliberately using the JavaScript API, your error handling model will be based on such errors.

Handling native/script errors is described in Error Handling, and the error message listing is in Silverlight Plug-in Error Messages.

7.3. Errors from Scripts, Tools, or Dialogs

The remainder of reference topics in this reference section describe errors that originate with specific scripts, tools, or tool dialogs that support Silverlight development.

7.4. Silverlight Plug-in Error Messages

This topic lists the error messages that come from the native code implementation of the Silverlight plug-in, and the native code level of the object tree and the XAML parser. The error messages are mainly relevant if you use the JavaScript API technique for error handling (OnError) or for general script-level debugging during development.

Note:

Handling errors for the JavaScript API is described in the topic Error Handling.

The error messages might not be relevant to managed user code exception handling with the UnhandledException event. However, in many cases, native errors can occur when you make managed calls to members that use functionality in the core. The native errors might be useful debugging clues during the development phase to determine what aspect of the managed call is not working as expected.

The following table lists the error codes and corresponding error messages. The error messages are displayed only when you have the Silverlight developer runtime installed. For more information, see Silverlight Tools. The end-user runtime provides error codes only, but you can use the following table to look up the correspond error message.

Error Code	Error Message
1001	AG_E_UNKNOWN_ERROR
2000	Document is not well-formed XAML.
2001	DTDs are not allowed.
2002	Entity references to named entities other than lt, gt, amp, apos, and quot are not allowed.
2003	Entity definitions are not allowed.
2004	Only UTF-8 and UTF-16 encodings are supported.
2005	Unknown namespace %0.
2006	http://schemas.microsoft.com/winfx/2006/xaml/presentation is not supported, did you mean http://schemas.microsoft.com/client/2007?

2007 Unknown element: %0.
2008 Cannot specify the value multiple times for properties.
2009 Cannot specify content for %0 and set its %1 property at the same time.
2010 %0 does not support %1 as content.
2011 %0 does not support text content.
2012 Unknown attribute %0 on element %1.
2013 Attribute %0 value is out of range.
2014 The attribute %0 is read only and cannot be set.
2015 Cannot declare custom namespaces in JavaScript.
2016 Cannot specify both Name and x:Name attributes.
2017 Null is not a legal value for attribute %0.
2018 The element %0 does not support attributes.
2019 Xaml may not use base type names, use %0 instead.
2020 Invalid Property: %0.
2021 Invalid enum value.
2022 Unknown Color: %0.
2023 %0 is not an event.
2024 Invalid attribute value %0 for property %1.
2025 AG_E_PARSER_ATTRIBUTE
2027 AG_E_PARSER_ELEMENT
2028 The name already exists in the tree: %0.
2030 Property element %0 cannot be used inside another property element.
2031 The property element %0 must appear before other content.
2033 Cannot specify the value multiple times for property: %0.
2034 Elements in a ResourceDictionary must have x:Key or x:Name attribute.
2035 The element is not valid in the given namespace.
2036 XamlReader.Load() does not accept event handlers.
2100 AG_E_INIT_CALLBACK

2101 Failed to initialize the application's root visual
2102 Invalid source file specified on Silverlight plug-in. Expected Uri for application file
2103 Invalid or malformed application: Check manifest
2104 Could not download the Silverlight application. Check web server settings
2105 Failed to load pre-requisites for the application
2106 Failed to load the application. It was built with an obsolete version of Silverlight
2107 Splash screen file needs to be on the same domain as the application
2108 Failed to download the splash screen
2201 AG_E_RUNTIME_INVALID_CALL
2202 AG_E_RUNTIME_FINDNAME
2203 AG_E_RUNTIME_SETVALUE
2204 AG_E_RUNTIME_GETVALUE
2205 AG_E_RUNTIME_ADDEVENT
2206 AG_E_RUNTIME_DELEVENT
2207 AG_E_RUNTIME_METHOD
2208 AG_E_RUNTIME_GETHOST
2209 AG_E_RUNTIME_GETPARENT
2210 AG_E_INVALID_ARGUMENT
2211 AG_E_RUNTIME_HTML_ACCESS_RESTRICTED
2212 Animation target not specified.
2213 Cannot resolve TargetName %0.
2214 Cannot resolve TargetProperty %0 on specified object.
2215 %0 cannot be used to animate property %1 due to incompatible type.
2216 Multiple animations in the same containing Storyboard cannot target the same property on a single element.
2217 KeyTime property on KeyFrame object must be set to a non-negative TimeSpan value.
2218 Operation is not valid on an active Animation or Storyboard. Root Storyboard must be stopped first.
2219 Operation is not allowed on a non-root Storyboard.
2220 Cannot determine Storyboard Duration.

2250 AG_E_RUNTIME_MANAGED_UNKNOWN_ERROR
2251 AG_E_RUNTIME_MANAGED_ACTIVATION
2252 AG_E_RUNTIME_MANAGED_ASSEMBLY_DOWNLOAD
2253 AG_E_RUNTIME_MANAGED_ASSEMBLY_LOAD
2254 AG_E_PARSER_UNKNOWN_TYPE
2255 AG_E_PARSER_BAD_TYPE
2256 AG_E_PARSER_BAD_NATIVE_TYPE
2257 AG_E_PARSER_CREATE_OBJECT_FAILED
2258 AG_E_PARSER_PROPERTY_NOT_FOUND
2259 AG_E_PARSER_BAD_PROPERTY_TYPE
2260 AG_E_PARSER_BAD_PROPERTY_VALUE
2261 AG_E_PARSER_ROOT_NOT_CUSTOM
2262 AG_E_PARSER_NAMESPACE_NOT_SUPPORTED
2263 AG_E_PARSER_MISSING_DEFAULT_NAMESPACE
2264 AG_E_PARSER_INVALID_XMLNS
2265 AG_E_PARSER_INVALID_CLASS
2266 AG_E_RUNTIME_MANAGED_BAD_DLR_SCRIPT
2267 AG_E_PARSER_INVALID_XCODE_TAG
2268 AG_E_PARSER_ELEMENT_IGNORED
2269 Elements in a ResourceDictionary must not set both x:Name and x:Key, or x:Key twice
2270 Attribute x:Key set on an invalid object, only objects in a ResourceDictionary can have x:Key
2271 Improper syntax for Static Resource (ie: {StaticResource Key})
2272 Cannot find a Resource with the Name/Key %0
2273 Elements in the same ResourceDictionary cannot have the same x:Key
2274 AG_E_RUNTIME_MANAGED_ASSEMBLY_NOT_FOUND
2275 Invalid characters in Source Uri.
3000 AG_E_UNABLE_TO_PLAY
3001 AG_E_INVALID_FILE_FORMAT

3002	AG_E_NOT_FOUND
3003	AG_E_MEDIA_DISCONNECTED
3004	Invalid ASX element
3005	Invalid ASX attribute
3006	Unsupported ASX element
3007	Unsupported ASX attribute
3008	ASX parse error
4000	AG_E_ABORT_FAILED
4001	AG_E_NETWORK_ERROR
4002	AG_E_ACCESS_DENIED
4003	AG_E_INVALID_POLICY
4004	AG_E_MANAGED_EXCEPTION
4005	XamlReader.Load() does not support x:Class.
4006	Only the root element can have an x:Class attribute.
4007	The root element must be <UserControl>, not a subclass of UserControl.
4008	Layout cycle detected. Layout could not complete.
4009	Element is already the child of another element.
4010	The ItemsControl.ItemsPanelTemplate must have a derivative of Panel as the root element.
4011	Setting the Template property on a UserControl is not supported.
4012	Cannot explicitly modify Children collection of Panel used as ItemsPanel for ItemsControl. ItemsControl generates child elements for Panel.
6000	Unable to play protected content
6001	Unable to individualize due to a network or protocol error
6002	Unable to acquire license due to a network or protocol error
6003	Unable to individualize due to an installation failure
6004	Update of Silverlight required
6005	Incorrect header format for protected media file
6006	Unable to process license

6007	Unable to acquire license due to an application specific error
6008	Individualization disabled by user
7000	unexpected end of input
7001	unrecognized encoding
7002	unable to switch the encoding
7003	unrecognized input signature
7004	whitespace expected
7005	semicolon expected
7006	'>' expected
7007	quote expected
7008	equal expected
7009	wfc: no '<' in attribute value
7010	hexadecimal digit expected
7011	decimal digit expected
7012	'[' expected
7013	'(' expected
7014	illegal xml character
7015	illegal name character
7016	incorrect document syntax
7017	incorrect CDATA section syntax
7018	incorrect comment syntax
7019	incorrect conditional section syntax
7020	incorrect ATTLIST declaration syntax
7021	incorrect DOCTYPE declaration syntax
7022	incorrect ELEMENT declaration syntax
7023	incorrect ENTITY declaration syntax
7024	incorrect NOTATION declaration syntax
7025	NDATA expected

7026 PUBLIC expected
7027 SYSTEM expected
7028 name expected
7029 one root element
7030 wfc: element type match
7031 wfc: unique attribute spec
7032 text/xmldecl not at the beginning of input
7033 leading \'xml\'
7034 incorrect text declaration syntax
7035 incorrect xml declaration syntax
7036 incorrect encoding name syntax
7037 incorrect public identifier syntax
7038 wfc: pes in internal subset
7039 wfc: pes between declarations
7040 wfc: no recursion
7041 entity content not well formed
7042 wfc: undeclared entity
7043 wfc: parsed entity
7044 wfc: no external entity references
7045 incorrect processing instruction syntax
7046 incorrect system identifier syntax
7047 '?' expected
7048 no ']]>' in element content
7049 not all chunks of value have been read
7050 DTD was found but is prohibited
7051 illegal qualified name character
7052 multiple colons in qualified name
7053 colon in name

7054 declared prefix
7055 undeclared prefix
7056 non default namespace with empty uri
7057 'xml' prefix is reserved and must have the http://www.w3.org/XML/1998/namespace URI
7058 'xmlns' prefix is reserved for use by XML
7059 xml namespace URI (http://www.w3.org/XML/1998/namespace) must be assigned only to prefix 'xml'
7060 xmlns namespace URI (http://www.w3.org/2000/xmlns/) is reserved and must not be used
7061 invalid decimal
7062 invalid hexidecimal
7063 invalid unicode
7064 invalid encoding
8001 Upgrade required
8002 Browser restart required

7.5. Warning: Web services at '<URL>' might not be accessible due to cross-domain restrictions

A crossdomain.xml file has been detected at the specified URL, but the required attribute for cross-domain access was not found.

If a Silverlight version 2 application needs to access a service in another domain, the service must explicitly opt in to enable cross-domain access. By opting in, a service affirms that the operations it exposes can safely be invoked by a Silverlight 2 control, without potentially damaging consequences to the data that the service stores.

Note:

To correct this error, you must have permissions to modify files at the root of the domain in which the service is hosted. You should also understand the security implications of enabling cross-domain access to the service.

To correct this error

1. Create a crossdomain.xml file that contains the following configuration. The file must be configured to enable access to the service from any other domain, or it is not recognized by Silverlight.

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <allow-access-from domain="*" />
</cross-domain-policy>
```

2. Save the crossdomain.xml file to the root of the domain in which the service is hosted. If, for example, the service is hosted on http://fabrikam.com, the file must be located at http://fabrikam.com/crossdomain.xml.
3. Test that the service is enabled by invoking it from your application.

7.6. Warning: Request for cross-domain policy at '<URL>' timed out

The connection to the specified URL timed out when the application was searching for a cross-domain policy file.

If a Silverlight version 2 application needs to access a service in another domain, the service must explicitly opt in to enable cross-domain access. By opting in, a service affirms that the operations it exposes can safely be invoked by a Silverlight 2 control, without potentially damaging consequences to the data that the service stores.

To correct this error

- Verify that your Internet connection is working by browsing to a different URL.
- Attempt to update the service reference. In Solution Explorer, right-click the service reference and then click Update Service Reference.

  A progress dialog box is displayed when the reference is being updated from its original location, and the service client is regenerated to reflect any changes in the metadata.

- Verify that either a clientaccesspolicy.xml file or a valid crossdomain.xml file exists at the root of the domain in which the service is hosted.

7.7. Warning: Could not locate cross-domain policy at '<URL>'

A cross-domain policy file for the service was not found at the specified URL.

If a Silverlight version 2 application needs to access a service in another domain, the service must explicitly opt in to enable cross-domain access. By opting in, a service affirms that the operations it exposes can safely be invoked by a Silverlight 2 control, without potentially damaging consequences to the data that the service stores.

To use a clientaccesspolicy.xml file to enable cross-domain access

1. Create a clientaccesspolicy.xml file that enables access to the service. The following configuration enables access from any other domain to all resources on the domain in which the clientaccesspolicy.xml file is located.

```
<?xml version="1.0" encoding="utf-8"?>
<access-policy>
  <cross-domain-access>
    <policy>
      <allow-from>
        <domain uri="*"/>
      </allow-from>
      <grant-to>
        <resource path="/" include-subpaths="true"/>
      </grant-to>
    </policy>
  </cross-domain-access>
</access-policy>
```

2. Alternatively, if you want to enable access form only one other domain, such as http://contoso.com, the clientaccesspolicy.xml should contain the following configuration.

```
<?xml version="1.0" encoding="utf-8"?>
<access-policy>
  <cross-domain-access>
    <policy>
      <allow-from>
        <domain uri="http://contoso.com"/>
      </allow-from>
      <grant-to>
        <resource path="/" include-subpaths="true"/>
      </grant-to>
    </policy>
  </cross-domain-access>
</access-policy>
```

3. Save the clientaccesspolicy.xml file to the root of the domain in which the service is hosted. If, for example, the service is hosted on http://fabrikam.com, the file must be located at http://fabrikam.com/clientaccesspolicy.xml.
4. Test that the access is enabled by invoking the service from your application.

To use a crossdomain.xml file to enable cross-domain access

1. Create a crossdomain.xml file that contains the following configuration. The file must be configured to enable access to the service from any other domain, or it is not recognized by Silverlight 2.

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM _
"http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <allow-access-from domain="*" />
</cross-domain-policy>
```

2. Save the crossdomain.xml file to the root of the domain in which the service is hosted. If, for example, the service is hosted on http://fabrikam.com, the file must be located at http://fabrikam.com/crossdomain.xml.
3. Test that the service is enabled by invoking the service from your application.

8. Silverlight Plug-in Object Reference

The Silverlight plug-in is an object that can be used with several programming models and access models. Members of the plug-in class can be accessed through the HTML DOM and through the object element object that is defined in order to host the Silverlight plug-in on an HTML page. The same functionality can also be accessed through APIs that produce the HTML page object tag as their output, such as the ASP.NET Silverlight and SilverlightPlugin Web server controls and the functions in the Silverlight.js library. Some Silverlight plug-in functionality can be accessed at run time by using one of the Silverlight programming models (the JavaScript API or managed code).

The topics in this section describe the core functionality of the Silverlight plug-in object. They provide the syntax for each programming model and access model. If functionality is available to only some access models, that information is noted in the topic.

The topic name is based on the name of the property or parameter for the object element. Some topics describe functionality that is not available for the object element as described here. In these cases, the topics are included because they describe ways to program the Silverlight plug-in object directly.

Note the following about the topics in this section:

- For the JavaScript syntax, the placeholder silverlightObject is used. You can get a reference to the silverlightObject object from the JavaScript API by calling GetHost on most Silverlight objects. Alternatively, you can get the silverlightObject instance from the DOM by calling Document.GetElementByID and specifying the ID of the object element that instantiates the Silverlight plug-in.
- The Silverlight.js syntax is based on the Silverlight.js functions that are current as of the release of this documentation. The Silverlight.js library might be updated by Microsoft. If so, the syntax might change, which includes parameter names and parameter order.
- For the COM syntax, the relevant interface and member are indicated. However, for more information, see COM Reference.

8.1. Property Values of the HTML Object Element

This topic describes properties of the DOM-level object element that are relevant for programming the Silverlight plug-in.

Property Values of the Object Tag

The following is a list of the properties of the object element that are relevant to Silverlight features, to Silverlight-based applications, or to APIs that are used to program Silverlight-based applications.

Note:

This list does not include properties of the object element that are not relevant to the Silverlight plug-in. For more information about additional properties of the object element, see the Document Object Model (DOM) Level 1 Specification on the W3C Web site.

Property	Description
classid, codebase, codetype, code, archive	Not typically used for a Silverlight plug-in object element. Silverlight uses type to specify a MIME type for its codebase.
type	Identifies the Silverlight plug-in codebase. This should be set to the following value: application/x-silverlight-2
data	Not used by most browsers for Silverlight purposes. However, setting a value for this property can help prevent decreased performance on some browsers. For this reason, the Silverlight project templates include the following value for the data property: data:application/x-silverlight, **Note:** The trailing comma is part of the property value.
width	Specifies the initial width of the Silverlight plug-in area in the HTML page. Can be as a pixel value or as a percentage (a value that ends with the % character specifies a percentage value). For example, "400" specifies 400 pixels, and "50%" specifies 50% (half) of the available width of the browser content area. For more information, see the Remarks section of this topic.
height	Specifies the initial height of the Silverlight plug-in area in the HTML page. Can be set either as a pixel value or a percentage (a value that ends with the % character specifies a percentage value). For example, "300" specifies 300 pixels, and "50%" specifies 50% (half) of the available height of the browser content area. For more information, see the Remarks section of this topic.
altHtml	Provides content that provides fallback cues such as download and installation information for Silverlight. We recommend that you do not specify altHtml as an attribute. Instead, provide this value as inner HTML for the object element. This technique is equivalent to providing an altHtml attribute value and gives you HTML that is easier to read.
tabIndex	Specifies the tab index for the object element. This attribute is compatible with a Silverlight object element. However, Silverlight has an internal accessibility model that incorporates a tab sequence.
id, style, etc.	Specify general behavior of the object element. These attributes are possible on a Silverlight object element, depending on how you construct the hosting HTML page and on the overall HTML support.
culture, uiculture	Specifies the culture settings that are used to retrieve satellite assemblies. For more information, see Localizing Silverlight

Applications.

Remarks

The height and width properties of the Silverlight object element determine the content area that is available for your application in the browser host. If you set height and width to a value that is smaller than any height and width values in your Silverlight page (such as the Height and Width of the UserControl root for a managed Silverlight page), the content might be clipped. If you leave Height and Width unspecified on the UserControl object, the page is sized to the size that is determined by the browser and by the height and width setting of the Silverlight object element.

8.2. ActualHeight (Silverlight Plug-in Object)

Gets the height of the rendering area of the Silverlight plug-in.

Syntax

object Element

Not applicable. This property is not settable on instantiation. However, the height property of the object element is settable and affects the ActualHeight value.

Silverlight.js

Not applicable. This property is not settable on instantiation. However, the height parameter of CreateObject and CreateObjectEx functions is settable and affects the ActualHeight value.

JaavaScript

```
value = silverlightObject.content.ActualHeight;
```

COM

IXcpControl::Content gets an IDispatch interface, which you can use to call ActualHeight.

Managed Code

See Content..::.ActualHeight.

Not available. This is usually determined by the value of the SilverlightPluginHeight()()() property.

Property Value

The height of the rendering area of the Silverlight plug-in, in pixels.

This property is read-only for all means of access. The default value is the height property value of the Silverlight plug-in, as specified by the object element that instantiated it.

Remarks

Important Note:

In the early stages of the object lifetime of a Silverlight plug-in, ActualHeight and ActualWidth do not contain usable values. In particular, when the OnLoad event occurs, correct values for ActualHeight and ActualWidth are not guaranteed. In general, you should check these values in the handler for the OnResize event, which occurs just after the OnLoad event when a Silverlight plug-in is first instantiated on an HTML page. If OnResize occurs again, it indicates that ActualHeight or ActualWidth have changed at run time, and any saved values should be adjusted.

A Silverlight plug-in can be displayed in either embedded mode or full-screen mode:

- In embedded mode, the plug-in is displayed in the Web browser content area.
- In full-screen mode, the plug-in is displayed on top of all other applications.

When the Silverlight plug-in is in embedded mode, the value of ActualHeight depends on whether the height of the Silverlight plug-in is a pixel count or a percentage value. The height of the Silverlight plug-in is defined in the object element of the containing HTML page. (The actual object element and its height might be specified indirectly, through Silverlight.js helper functions or through an ASP.NET control.) If the height is a pixel-count value, such as "300", ActualHeight is set to the plug-in height. If the height of the Silverlight plug-in is a percentage value, such as "100%", ActualHeight is set to the most appropriate height for displaying the plug-in in a browser. In this case, if the browser window height is changed (either through a user action or by manipulating the HTML DOM), the ActualHeight value changes also.

When the application is in full-screen mode, the value of ActualHeight is set to the current vertical resolution of the user's display. For more information, see Full-Screen Support.

When the Silverlight plug-in is in embedded mode, the OnResize event occurs whenever the calculated values of the ActualHeight and ActualWidth properties change. When the Silverlight plug-in is in full-screen mode, the OnResize event does not occur. If the mode changes from embedded to full-screen, or full-screen to embedded, the OnResize event does not occur. Instead, the OnFullScreenChanged event occurs.

8.3. ActualWidth (Silverlight Plug-in Object)

Gets the width of the rendering area of the Silverlight plug-in.

⊟ Syntax

object Element

Not applicable. This property is not settable on instantiation. However, the width property of the object element is settable and affects the ActualWidth value.

Silverlight.js

Not applicable. This property is not settable on instantiation. However, the width parameter of the CreateObject and CreateObjectX functions is settable and affects the ActualWidth value.

JavaScript

```
value = silverlightObject.content.ActualWidth;
```

COM

IXcpControl::Content gets an IDispatch interface, which you can use to call ActualWidth.

Managed Code

See Content..::.ActualWidth.

ASP.NET Controls for Silverlight Client

Not available. This is usually determined by the value of the SilverlightPluginWidth()()() property.

⊟ Property Value

The width of the rendering area of the Silverlight plug-in, in pixels.

This property is read-only for all means of access. The default value is the height property value of the Silverlight plug-in, as specified by the object element that instantiated it. If height was specified by percentage during initialization, this is a calculated pixel value.

Remarks

Important Note:

In the early stages of the object lifetime of a Silverlight plug-in, ActualHeight and ActualWidth do not contain usable values. In particular, when the OnLoad event occurs, correct values for ActualHeight and ActualWidth are not guaranteed. In general, you should check these values in the handler for the OnResize event, which occurs just after OnLoad whenever a Silverlight plug-in is first instantiated on an HTML page. If OnResize occurs again, it indicates that ActualHeight or ActualWidth have changed at run time, and any saved values should be adjusted.

A Silverlight plug-in can be displayed in either embedded mode or full-screen mode:

- In embedded mode, the plug-in is displayed in the Web browser content area.
- In full-screen mode, the plug-in is displayed on top of all other applications.

When the Silverlight plug-in is in embedded mode, the value of ActualWidth depends on whether the height of the Silverlight plug-in is a pixel count or a percentage value. The width of the Silverlight plug-in is defined in the object element of the containing HTML page. (The actual object element and its width might be specified indirectly, through Silverlight.js helper functions or through an ASP.NET control.) If the width is a pixel count value, such as 300, ActualHeight is set to the plug-in width. If the width of the Silverlight plug-in is a percentage value, such as "100%", ActualWidtht is set to the most appropriate width for displaying the plug-in in a browser. In this case, if the browser window height is changed (either through a user action or by manipulating the HTML DOM), the ActualWidth value changes also.

When the application is in full-screen mode, the value of ActualWidth is set to the current horizontal resolution of the user's display. For more information, see Full-Screen Support.

The OnResize event occurs whenever the calculated values of the ActualHeight and ActualWidth properties change, and the Silverlight plug-in is in embedded mode. When the Silverlight plug-in is in full-screen mode, the OnResize event does not occur. If the mode changes (from embedded to full-screen, or full-screen to embedded) then the OnResize event does not occur. The OnFullScreenChanged event occurs instead.

8.4. AllowHtmlPopupWindow (Silverlight Plug-in Object)

Gets or sets a value that indicates whether the hosted content in the Silverlight plug-in can use the HtmlPage..::.PopupWindow method to display a new browser window.

Syntax

object Element

```
<object ...>
  <param name="allowHtmlPopupWindow" value="bool"/>
  ...
</object>
```

Silverlight.js

```
Silverlight.CreateObject(,,,{allowHtmlPopupWindow:'bool'});
-or-
Silverlight.CreateObjectEx({properties:{allowHtmlPopupWindow:'bool'}});
```

JavaScript

Not available.

COM

Not available.

Managed Code

See HtmlPage..::.IsPopupWindowAllowed.

ASP.NET Controls for Silverlight Client

Not available.

Property Value

true if the hosted content can use the HtmlPage..::.PopupWindow method to display a new browser window; otherwise, false. The default is true for same-domain applications and false for cross-domain applications.

Remarks

This property can only be set during Silverlight plug-in initialization, and is read-only at run time for all possible access models.

You can set this property to true in cross-domain settings to bypass browser-based popup blockers.

Caution:

When a host page enables HTML access, it gives the Silverlight-based application complete access to its DOM, including the ability to modify content, insert and evaluate JavaScript code, and so on. This should be allowed only if the host page trusts the Silverlight-based application. By default, HTML access is enabled for same-domain Silverlight-based applications because there is implicit trust between HTML and same-domain Silverlight content. However, access must be specifically enabled for cross-domain applications. For more information, see Security Settings in HTML Bridge.

In the JavaScript API, you always have access to the HTML DOM. If you use the JavaScript API exclusively, the value of this property is ignored.

8.5. AutoUpgrade (Silverlight Plug-in Object)

Sets a value that indicates whether a Silverlight plug-in version earlier than the specified minRuntimeVersion will attempt to update automatically.

Syntax

object Element

```
<object ...>
  <param name="autoUpgrade" value="bool"/>
  ...
</object>
```

Silverlight.js

```
Silverlight.CreateObject(,,,{autoUpgrade:'bool'});
-or-
Silverlight.CreateObjectEx({properties:{autoUpgrade:'bool'}})
```

JavaScript

Not available.

COM

Not available.

Managed Code

Not available.

ASP.NET Controls for Silverlight Client

See SilverlightAutoUpgrade()()() and SilverlightPluginAutoUpgrade()()() (server controls).

⊟ Property Value

true if a Silverlight plug-in version earlier than minRuntimeVersion should attempt to update automatically; otherwise, false. The default is true.

⊟ Remarks

Setting autoUpgrade has no effect unless you also set minRuntimeVersion.

If an upgrade is required and you set autoUpgrade to true, the plug-in will prompt the user to upgrade and then to restart the Web browser.

If an upgrade is required, the following errors occur, regardless of the autoUpgrade setting:

- 8001 - Upgrade required.
- 8002 - Browser restart required.

You can respond to these errors in an OnError handler in order to display your own upgrade and browser restart messages. In this case, however, you typically do not set autoUpgrade to true.

8.6. Background (Silverlight Plug-in Object)

Gets or sets the background color of the rectangular region that displays XAML content.

Syntax

object Element

```
<object ...>
  <param name="background" value="colorValue"/>
  ...
</object>
```

Silverlight.js

```
Silverlight.CreateObject(,,,{background:'colorValue'});
-or-
Silverlight.CreateObjectEx({properties:{background:'colorValue'}});
```

JavaScript

```
            value = silverlightObject.settings.Background;
silverlightObject.settings.Background = colorValue;
```

COM

IXcpControl::Content gets an IDispatch interface, which you can use to call Background.

Managed Code

See SilverlightHost..::.Background. In managed code, this property is read-only at run time.

ASP.NET Controls for Silverlight Client

See SilverlightPluginBackground()()().

Property Value

Specifies the background color value as a string. This string can specify a named color value or a color value that is specified by using several possible conventions, which includes RGB and ScRGB. RGB and ScRGB values can be specified with or without alpha transparency. The following table lists conventions that are used to specify color values.

Convention	Example	Description

Named color	Red	A named color value. (The name is not case-sensitive.) For more information about color names, see Color.
8-bit RGB, no alpha	#F00	An 8-bit RGB value, in the form #RGB.
8-bit RGB, alpha	#8F00	An 8-bit RGB value with alpha transparency, in the form #ARGB.
16-bit RGB, no alpha	#FF0000	A 16-bit RGB value, in the form #RRGGBB.
16-bit RGB, alpha	#80FF0000	A 16-bit RGB value with alpha transparency, in the form #AARRGGBB.
scRGB, no alpha	sc#1,0,0	An ScRGB value without alpha transparency, in the form sc#R,G,B.
scRGB, alpha	sc#0.5,1,0,0	An ScRGB value without alpha transparency in the form sc#A,R,G,B.

The default value is null.

⊟ Remarks

For rendering purposes, a null value for the Background property of the Silverlight plug-in is interpreted as the named color White.

If the Windowless property is set to true, or if Silverlight is running on a Macintosh computer (where windowless mode is always active, regardless of the Windowless setting), the Background property value can specify an alpha value for the color. The color value is then blended with the HTML page.

Important Note:

Setting an alpha channel value for full or partial transparency has possible performance consequences, especially in certain client environments .

8.7. EnableFramerateCounter (Silverlight Plug-in Object)

Gets or sets a value that indicates whether to display the current frame rate in the hosting browser's status bar.

⊟ Syntax

object Element

Not applicable. This property is not settable on instantiation.

Silverlight.js

Not applicable. This property is not settable on instantiation.

JavaScript

```
            silverlightObject.settings.EnableFramerateCounter = bool;
value = silverlightObject.settings.EnableFramerateCounter;
```

COM

IXcpControl::Settings get an IDispatch interface, which you can use to call EnableFramerateCounter.

Managed Code

See Settings..::.EnableFrameRateCounter.

ASP.NET Controls for Silverlight Client

See SilverlightEnableFrameRateCounter()()().

Remarks

This feature works only in Microsoft Internet Explorer on Windows XP Service Pack 2 or Windows Vista. Setting this value has no effect in other browsers.

The EnableFrameRateCounter property is useful as a diagnostic tool. You can use the frame-rate information to identify performance bottlenecks in your application.

This property cannot be set in an initial object element. You can set it at run time after the object is instantiated. For example, you can set it in a OnLoad event handler that is attached to the Silverlight plug-in.

The format of the frame rate counter is as follows:

fps:currentFramerate/maxFramerate

The currentFramerate value is the optimal frame rate that would apply if there is no an upper frame rate limit, based on conditions in the plug-in environment and as reported by the plug-in. The maxFramerate value is configurable through the maxframerate parameter of the Silverlight plug-in object. The maxframerate parameter defaults to 60. Each number is a value that reports frames per second (fps). The actual displayed frame rate is

set to whichever number is lower. You can illustrate the relationship between currentFramerate and maxFramerate by setting a deliberately low maxframerate value, such as 2 per second.

8.8. EnableHtmlAccess (Silverlight Plug-in Object)

Gets or sets a value that indicates whether the hosted content in the Silverlight plug-in and in the associated run-time code has access to the browser Document Object Model (DOM).

Syntax

object Element

```
<object ...>
  <param name="enablehtmlaccess" value="bool"/>
  ...
</object>
```

Silverlight.js

```
Silverlight.CreateObject(,,,{enableHtmlAccess:'bool'});
-or-
Silverlight.CreateObjectEx({properties:{enableHtmlAccess:'bool'}});
```

JavaScript

```
            bool = silverlightObject.settings.EnableHtmlAccess;
```

COM

Reported as a value of IXcpControlHost::GetHostOptions.

Managed Code

See Settings..::.EnableHTMLAccess.

ASP.NET Controls for Silverlight Client

See SilverlightHtmlAccess()()().

Property Value

true if the hosted content and associated code has access to the browser HTML DOM; otherwise, false. The default value is true for same-domain applications and false for cross-domain applications.

Remarks

This property can only be set during Silverlight plug-in initialization, and is read-only at run-time for all possible access models.

If EnableHtmlAccess is false and you are using the managed API, most of the APIs in the System.Windows.Browser namespace will raise an exception when they are called. For instance, when EnableHtmlAccess is set to false, getting the HtmlPage..::.BrowserInformation property value to determine browser host information (which ultimately comes from the HTML DOM) raises an InvalidOperationException exception. For more information, see HTML Bridge: Interaction Between HTML and Managed Code.

Caution:

When a host page enables HTML access, it allows the Silverlight-based application complete access to its DOM, including the ability to modify content, insert and evaluate JavaScript code, and so on. This should be allowed only if the host page trusts the Silverlight-based application. HTML access is enabled by default for same-domain Silverlight-based applications because there is implicit trust between HTML and same-domain Silverlight content. However, access must be specifically enabled for cross-domain applications. For more information, see Security Settings in HTML Bridge.

In the JavaScript API, you always have access to the HTML DOM. If you use the JavaScript API exclusively, the value of this property is ignored.

8.9. EnableRedrawRegions (Silverlight Plug-in Object)

Gets or sets a value that determines whether to visually indicate the areas of the plug-in that are being redrawn with each frame.

Syntax

object Element

Not applicable. This property is not settable on instantiation.

Silverlight.js

Not applicable. This property is not settable on instantiation.

JavaScript

```
            silverlightObject.settings.EnableRedrawRegions = bool;
value = silverlightObject.settings.EnableRedrawRegions;
```

COM

IXcpControl::Settings get an IDispatch interface, which you can use to call EnableRedrawRegions.

Managed Code

See Settings..::.EnableRedrawRegions.

ASP.NET Controls for Silverlight Client

See SilverlightEnableRedrawRegions()()().

Property Value

true if the areas of the plug-in that are being redrawn with each frame will be displayed; otherwise, false. The default value is false.

Remarks

The EnableRedrawRegions property is used during development to tune the performance of your application by showing the areas of the plug-in that are being redrawn with each frame. You cannot set this property during initialization. Typically you turn this property on and off by using an event handler. If you are using JavaScript, you typically do this in a handler for the OnLoad event. If you are using managed code, you typically do this in a handler for the Loaded event.

8.10. FullScreen (Silverlight Plug-in Object)

Gets or sets a value that indicates whether the Silverlight plug-in is displayed as a full-screen plug-in.

Syntax

object Element

Not applicable. This property is not settable on instantiation.

Silverlight.js

Not applicable. This property is not settable on instantiation.

JavaScript

```
            silverlightObject.content.fullScreen = bool;
bool = silverlightObject.content.fullScreen;
```

COM

Not available.

Managed Code

See Content..::.IsFullScreen.

ASP.NET Controls for Silverlight Client

Not available.

Property Value

true if the Silverlight plug-in is displayed as a full-screen plug-in; false if the Silverlight plug-in is displayed as an embedded plug-in. The default value is false.

Remarks

For more information about full-screen behavior, which includes limitations on input, see Full-Screen Support.

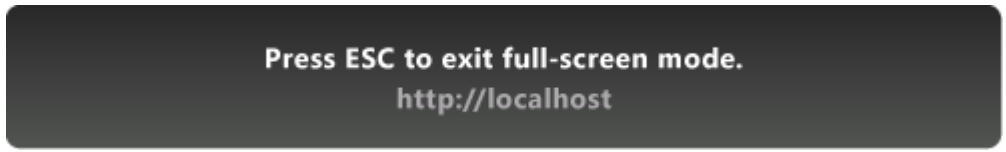
You cannot set this property during instantiation. You cannot set this value in an onLoad handler or Loaded handler. Typically, you switch to full-screen mode only in response to a user action. You can set IsFullScreen only in a handler for a user-initiated event, such as a mouse event or a keyboard event that is captured by the UI.

A Silverlight plug-in can be displayed in either embedded mode or full-screen mode:

- In embedded mode, the plug-in is displayed in the Web browser content area.
- In full-screen mode, the plug-in is displayed on top of all other applications.

When a Silverlight plug-in is displayed in full-screen mode, a message is displayed briefly. This message alerts the user that the application is in full-screen mode, and provides information about how to return to embedded mode. The following illustration shows this message.

Full-screen mode message (English)



The OnFullScreenChanged event occurs when the value of the FullScreen property changes.

Note:

For best performance, when your application goes into full-screen mode, hide or disconnect from the tree all the objects that are not being rendered in full-screen mode. You can hide an object by setting its Visibility property to Collapsed.

8.11. InitParams (Silverlight Plug-in Object)

Gets or sets user-defined initialization parameters.

Syntax

object Element

```
<object ...>
  <param name="initparams" value="initParamsString"/>
  ...
</object>
```

Silverlight.js

```
Silverlight.CreateObject(,,,,,"initParamsString");
-or-
Silverlight.CreateObjectEx({initparams:"initParamsString"});s
```

JavaScript

```
value = silverlightObject.InitParams;
```

COM

IXcpControl::InitParams

Managed Code

See StartupEventArgs..::.InitParams.

ASP.NET Controls for Silverlight Client

See SilverlightInitParameters()()().

Property Value

A string that represents a set of user-defined initialization parameters. The format of the value can be user-defined The default value is null or empty string.

Remarks

The InitParams property is write-once during instantiation, and read-only at run time. It cannot be set after initialization. The InitParams initialization parameter specifies user-defined parameter values when you create an instance of the Silverlight plug-in in an HTML page. InitParams provides a convenient way to pass custom string values from the HTML Document Object Model (DOM) to the Silverlight programming model.

Generally, if you have more than one string value to pass, you use a comma as a delimiter in the InitParams string. In the JavaScript API, you can write code to parse and split the string into separate parameters. If you use managed code, the StartupEventArgs..::.InitParams object in which the property value is passed expects a particular format for the input string. The InitParams values are provided as event data for the Startup event, instead of as a property of the SilverlightHost instance. In this case, the event data object contains the data as a dictionary. To be processed correctly into a dictionary, your string must have key-value pairs with an equals sign between key and value. For more than one key-value pair, use commas as delimiters between key=value pairs. For details, see StartupEventArgs..::.InitParams.

8.12. IsLoaded (Silverlight Plug-in Object)

Gets a value that indicates whether the Silverlight plug-in is loaded.

Syntax

object Element

Not applicable. This property is not settable on instantiation.

Silverlight.js

Not applicable. This property is not settable on instantiation.

JavaScript

```
silverlightObject.IsLoaded = bool;
```

COM

IXcpControl::IsLoaded

Managed Code

See SilverlightHost..::.IsLoaded.

ASP.NET Controls for Silverlight Client

Not available.

Property Value

true if the Silverlight plug-in is loaded; otherwise, false. The default value is false.

Remarks

The IsLoaded property is set to true after the XAML content in the Silverlight plug-in has completely loaded, but before the OnLoad event occurs.

8.13. MaxFrameRate (Silverlight Plug-in Object)

Gets or sets the maximum number of frames that Silverlight can render per second.

Syntax

object Element

```
<object ...>
  <param name="maxframerate" value="int"/>
  ...
</object>
```

Silverlight.js

```
Silverlight.CreateObject(,,,{maxFrameRate:'int'});
-or-
Silverlight.CreateObjectEx({properties:{maxFrameRate:'int'}});
```

JavaScript

```
            int = silverlightObject.settings.MaxFrameRate;
silverlightObject.settings.MaxFrameRate = int;
```

COM

IXcpControl::Settings gets an IDispatch interface, which you can use to call MaxFrameRate.

Managed Code

See Settings..::.MaxFrameRate.

ASP.NET Controls for Silverlight Client

See SilverlightMaxFrameRate()()().

Property Value

An integer value that specifies the maximum number of frames to render per second. The default value is 60.

Remarks

This property specifies the effective maximum frame rate for your application at run time. The actual frame rate that Silverlight content renders depends on system load performance and other factors.

Note that there is a mismatch in naming between the parameter in Silverlight.js functions and the other access models. The Silverlight.js parameter is named framerate, but the object element parameter and other properties are named MaxFrameRate. Both names refer to the same underlying property.

You can see the MaxFrameRate value by enabling a performance-tuning option that displays frames per second for Silverlight rendering. For more information, see EnableFramerateCounter (Silverlight Plug-in Object).

8.14. MinRuntimeVersion (Silverlight Plug-in Object)

Sets a value that indicates the earliest Silverlight version that is required in order to run a Silverlight-based application.

Syntax

object Element

```
<object ...>
  <param name="minRuntimeVersion" value="version"/>
  ...
</object>
```

Silverlight.js

```
Silverlight.CreateObject(,,,{minRuntimeVersion:'version'});
-or-
Silverlight.CreateObjectEx({properties:{minRuntimeVersion:'version'}})
```

JavaScript

Not available.

COM

Not available.

Managed Code

Not available.

ASP.NET Controls for Silverlight Client

See SilverlightMinimumVersion()()() and SilverlightPluginMinimumVersion()()() (ASP.NET Web server controls).

Property Value

The version of Silverlight that is required by the application. The default is the currently installed version, or null if Silverlight is not installed.

Remarks

The format of the version string is described in the IsVersionSupported topic.

If an upgrade is required, the Silverlight plug-in displays the message "Click to install the required version". Additionally, if you set the AutoUpgrade property to true, the plug-in will prompt the user to upgrade and then to restart the Web browser.

If an upgrade is required, the following errors occur, regardless of the AutoUpgrade setting:

- 8001 - Upgrade required.
- 8002 - Browser restart required.

You can respond to these errors in an OnError handler in order to display your own upgrade and browser restart messages.

8.15. OnError (Silverlight Plug-in Object)

Specifies the name of the handler to call when the Silverlight plug-in generates a parse error or run-time error at the native-code level.

Syntax

object Element

```
<object ...>
  <param name="onerror" value="functionname"/>
```

```
  ...
</object>
```

Silverlight.js

```
Silverlight.createObject(,,,{},{onError:handlername}};
-or-
Silverlight.CreateObjectEx({events:{onError:handlername});
```

JavaScript

You cannot assign the handler at run time. You can get the value, but the value is just the function name.

COM

IXcpControl::OnError.

Managed Code

Managed code has the first chance to catch errors from user code as exceptions. For more information, see Error Handling.

ASP.NET Controls for Silverlight Client

See SilverlightOnPluginError()()() (ASP.NET Web server control) or Sys.UI.Silverlight.Control.onPluginError Method (client framework)

Property Value

The name of the function that is invoked when the Silverlight plug-in generates a parse or run-time error at the native-code level. The default value is null.

Remarks

If you use the Silverlight.js client library, the Silverlight.js functions might define a default error handler and then reference it in the default behavior for CreateObject and CreateObjectEx. In that case, the default value of null might not apply. For more information, see Silverlight.js Reference.

Errors might be generated in the Silverlight run-time components in the following situations:

- During initial parsing of the XAML for a Silverlight-based application.
- When the JavaScript API is used and errors occur at run time.

Other handlers are available for asynchronous events that are associated with particular objects or actions. For example, you can handle the DownloadFailed event of the Downloader class. These events are not processed by an OnError handler.

Arguments for an OnError Event Handler Function

sender	object The Silverlight plug-in that invoked the event.
errorEventArgs	object The error and its source location. For more information, see ErrorEventArgs.

8.16. OnFullScreenChanged (Silverlight Plug-in Object)

Occurs whenever the FullScreen property of the Silverlight plug-in changes.

Syntax

object Element

```
<object ...>
  <param name="onfullscreenchanged" value="functionname"/>
  ...
</object>
```

Silverlight.js

Not available.

JavaScript

silverlightObject.Content.OnFullScreenChange = *handlername*;

Note:

For the object element, the parameter name is onfullscreenchanged. However, in the JavaScript API, you set the handler name by using the onfullscreenchange property (note the missing "d").

COM

Not available.

Managed Code

See Content..::.FullScreenChanged.

ASP.NET Controls for Silverlight Client

See SilverlightOnPluginFullScreenChanged()()() (ASP.NET Web server control) or Sys.UI.Silverlight.Control.onPluginFullScreenChanged Method (client framework).

Property Value

The name of the function that is invoked when the value of the FullScreen property changes. The default value is null.

Remarks

The OnFullScreenChanged event occurs whenever the Silverlight plug-in changes between embedded mode and full-screen mode. In embedded mode, the plug-in is displayed in the Web browser window. In full-screen mode, the plug-in is displayed on top of all other applications.

When the Silverlight plug-in is displayed in embedded mode, the plug-in is contained in the browser window. When the Silverlight plug-in is displayed in full-screen mode, the plug-in is resized to the current resolution of the display and overlays all other applications, including the browser.

For more information about full-screen behavior, see Full-Screen Support.

Arguments for an OnFullScreenChanged Event Handler Function

object
sender
The Silverlight plug-in that raised the event.

args Always null.

8.17. OnLoad (Silverlight Plug-in Object)

Occurs when the Silverlight plug-in has finished loading into the browser DOM.

Syntax

object Element

```
<object ...>
  <param name="onload" value="functionname"/>
  ...
</object>
```

Silverlight.js

```
Silverlight.createObject(,,,{},{onLoad:handlername}}
-or-
Silverlight.CreateObjectEx({events:{onLoad:handlername});
```

JavaScript

You can get the value, but the value is just the function name.

COM

IXcpControl::OnLoad

Managed Code

Not available directly. Most scenarios can be addressed by handling either the FrameworkElement..::.Loaded event from the page root, or by handling the Application..::.Startup event.

ASP.NET Controls for Silverlight Client

See OnPluginLoaded()()() (ASP.NET Web server control) or Sys.UI.Silverlight.Control.onPluginLoaded Method (client framework).

Remarks

The OnLoad event occurs after the content in the Silverlight plug-in has completely loaded. For managed code, the content is typically a .xap file that contains one or more XAML files. For the JavaScript API, the content is a single XAML file.

You cannot attach handlers to the OnLoad event after plug-in initialization, because a forced reload of a Silverlight plug-in does not preserve any state.

In general, do not make sizing decisions in a handler for the OnLoad event. The values of the ActualHeight and ActualWidth properties of the Silverlight plug-in are not guaranteed to be set at the time that the OnLoad event occurs. Instead, handle the OnResize event, which occurs when the ActualHeight or ActualWidth properties change, including when the plug-in first loads.

The OnLoad occurs after the Loaded event occurs for the root element of a page. The root element's Loaded event is in turn is guaranteed to occur after the Loaded event has occurred for all child elements under the root. This is true whether the content is using the JavaScript API or managed code.

The read-only IsLoaded property is set just before the OnLoad event occurs. This property indicates whether the Silverlight plug-in is loaded.

You can add handlers for the Silverlight.js version of the OnLoad event (for example, when you use Silverlight.createObject). The Silverlight.js library wraps the native plug-in implementation of the event. The wrapped version uses a different signature and a different model for an OnLoad event handler, which includes a context. For more information about the Silverlight.js version of OnLoad, see Silverlight.js Reference.

Arguments for an OnLoad Event Handler Function

object
sender
The Silverlight plug-in that raised the event.

args Always null.

8.18. OnResize (Silverlight Plug-in Object)

Syntax

object Element

```
<object ...>
  <param name="onresize" value="functionname"/>
  ...
</object>
```

Silverlight.js

Not available.

JavaScript

silverlightObject.Content.OnResize = *handlername*;

COM

Not available.

Managed Code

See Content..::.Resized.

ASP.NET Controls for Silverlight Client

See SilverlightOnPluginResized()()().

Remarks

The OnResize event occurs when the Silverlight plug-in is in embedded mode and the value of the ActualHeight or ActualWidth properties changes.

If a change in size of the Silverlight plug-in triggers the OnFullScreenChanged event, the OnResize event does not occur. The OnResize event also does not occur when the Silverlight plug-in is in full-screen mode.

In embedded mode, the plug-in displays in the Web browser window. In full-screen mode, the plug-in displays on top of all other applications.

Note:

Silverlight does not support automatic layout of elements based on changes to the plug-in size. Applications must provide resize logic for when the plug-in size changes.

Handle the OnResize event instead of the OnLoad event when you want to respond to changes to the plug-in size (as opposed to changes to the size of other elements in the HTML DOM). The values of the ActualHeight and ActualWidth properties of the Silverlight plug-in are not guaranteed to

be set at the time that the OnLoad event occurs. However, OnResize occurs when the values of the ActualHeight or ActualWidth properties change, which includes when the plug-in first loads.

Arguments for an OnResize Event Handler Function

object
sender
: The Silverlight plug-in that raised the event.

args
: Always null.

8.19. OnSourceDownloadComplete (Silverlight Plug-in Object)

Gets or sets the name of the event handler that is called when the source download has finished.

Syntax

object Element

```
<object ...>
  <param name="onsourcedownloadcomplete" value="functionname"/>
  ...
</object>
```

Silverlight.js

Not implemented in the functions.

JavaScript

Do not use; use the object parameter as the handler attachment point.

COM

IXcpControl::OnSourceDownloadComplete

Managed Code

Not available.

See SilverlightOnPluginSourceDownloadComplete()()() (ASP.NET Web server control) or Sys.UI.Silverlight.Control.pluginSourceDownloadComplete Event (client framework).

⊟ Property Value

The name of the function that is invoked when the source download has finished. The default value is null.

⊟ Remarks

The event reports download completion for the package that has been requested through the Source property.

The most typical scenario for noting download completion is to display a message in an area of the hosting HTML page (not in the splash screen). The splash-screen XAML page has reached the end of its lifetime when the SourceDownloadCompleted event occurs. Therefore, it does not make sense to report download completion in the splash screen.

Any JavaScript code for the splash screen (including event handlers that are referenced by this property) must be made in a script block or in an included script file that is referenced from the hosting HTML page.

For more information about how to work with splash screens, see Silverlight Splash Screens.

Arguments for an OnSourceDownloadCompleted Event Handler Function

sender	object The Silverlight plug-in that raised the event.
args	Always null.

8.20. OnSourceDownloadProgressChanged (Silverlight Plug-in Object)

Gets or sets the name of the event handler that is called when the source download progress changes.

⊟ Syntax

object Element

```
<object ...>
  <param name="onsourcedownloadprogresschanged" value="functionname"/>
  ...
</object>
```

Silverlight.js

Not implemented in the functions.

JavaScript

Do not use; use the object parameter as the handler attachment point.

COM

IXcpControl::OnSourceDownloadProgressChanged

Managed Code

Not available.

ASP.NET Controls for Silverlight Client

See SilverlightOnPluginSourceDownloadProgressChanged()()() (server control) or Sys.UI.Silverlight.Control.pluginSourceDownloadProgressChanged Event (client framework).

Property Value

The name of the function that is invoked when the source download progress changes. The default value is null.

Remarks

You can handle this event in order to report download progress in a splash screen or in the hosting HTML page.

The event reports progress for the package that has been requested through the Source property. The event that reports download progress is raised at intervals that are determined internally based on the size and the time of the download. The event is not raised as soon a particular percentage increment is downloaded. For short downloads, the event might be raised only a few times. For long downloads, the event might be raised thousands of times.

The most typical scenario for tracking download progress is to display the progress information to the user in a splash screen. The splash-screen XAML page must always use the JavaScript API. Any JavaScript code for the splash screen (including event handlers that are referenced by this property) must be made in a script block or in an included script file that is referenced from the hosting HTML page. You cannot handle OnSourceDownloadProgressChanged from managed code.

For more information about how to work with splash screens, see Silverlight Splash Screens.

Arguments for an OnSourceDownloadProgressChanged Event Handler Function

sender	object The Silverlight plug-in that raised the event.
downloadProgressEventArgs	object An object that contains information about download progress. The Progress property of this object contains a number between 0 and 1, which indicates how much of the document has been downloaded.

8.21. Source (Silverlight Plug-in Object)

Gets or sets the Uniform Resource Identifier (URI) of either the initial XAML file (JavaScript API) or XAP package (managed API) that specifies or contains the content to render.

Syntax

object Element

```
<object ...>
  <param name="source" value="sourceUri"/>
  ...
</object>
```

Silverlight.js

```
Silverlight.CreateObject('sourceUri');
-or-
Silverlight.CreateObjectEx({source:'sourceUri'})
```

JavaScript

```
            value = silverlightObject.Source;
silverlightObject.Source = sourceUri;
```

COM

IXcpControl::Source

Managed Code

See SilverlightHost..::.Source. This property is read-only for managed code.

ASP.NET Controls for Silverlight Client

See SilverlightSource()()() (ASP.NET Web server control) or Sys.UI.Silverlight.Control.source Property (client framework).

Property Value

The URI of the content or package to load into the Silverlight plug-in. The default is null.

Remarks

Changing the Source value at run time is permitted, under either the JavaScript API or the HTML DOM. This can affect performance. For the JavaScript API, this operation is equivalent to removing and adding the object element for the Silverlight plug-in using the DOM. If you reset Source, all state from the previous instance of the Silverlight plug-in is lost. Calling the CreateFromXaml method and rebuilding the object tree below the original root can provide better performance than resetting the Source property.

Changing the Source value at run time is not enabled through any managed API. You must use the DOM or manipulate the hosting HTML page in order to change the Source value for a managed-API application.

The string value of the Source property is interpreted as a URI instead of as an Internationalized Resource Identifier (IRI). Characters that are used to identify a Source value that are outside the US-ASCII character set must be encoded.

A Source URI can represent a relative path. The relative path is based on the location of the HTML page that contains the object element. If you use a relative path, always use the slash character (/) for traversal. Silverlight does not support the backslash character (\) for relative paths.

For security reasons, you cannot set the Source property to a URI that is not from the site of origin of the HTML page that hosts the Silverlight plug-in instance. Therefore, you should you should use only relative references to specify a Source URI.

The # (number sign) prefix in the string specifyies a Source value that references inline XAML. However, inline XAML is supported only for the JavaScript API. See Using Inline XAML.

8.22. SplashScreenSource (Silverlight Plug-in Object)

Gets or sets the URI of the splash screen XAML page. The splash screen displays while the Silverlight core components and the content package for a managed API application are loading.

Syntax

object Element

```
<object ...>
  <param name="splashscreensource" value="uri"/>
  ...
</object>
```

Silverlight.js

Not implemented in Silverlight.js.

JavaScript

Do not use; use the object parameter as the handler attachment point.

COM

IXcpControl::SplashScreenSource

Managed Code

Not available.

ASP.NET Controls for Silverlight Client

See SilverlightSplashScreenSource()()() (ASP.NET Web server control) or Sys.UI.Silverlight.Control.splashScreenSource Property (client framework).

⊟ Property Value

The URI of the content to load as the splash screen source. The default is null.

⊟ Remarks

The splash screen for a Silverlight-based application is generally used only if the main body of the application uses the managed API. Typically, you implement a splash screen only if the main body of the application takes enough time to load to justify displaying the splash screen.

The splash screen XAML page must always use the JavaScript API. Any JavaScript code for the splash screen must in a script block, or in an included script file that is referenced from the hosting HTML page.

For security reasons, the splash screen XAML page must be served from the same domain as the hosting HTML page for the Silverlight-based application. It must also be from the same domain as the document that is referenced in the Source property.

A SplashSceenSource URI can represent a relative path. The relative path is based on the location of the HTML page that contains the object element. If you use a relative path, always use the slash character (/) for traversal. Silverlight does not support the backslash character (\) for relative paths.

For more information about how to use splash screens, see Silverlight Splash Screens.

8.23. Windowless (Silverlight Plug-in Object)

Gets or sets a value that determines whether the Silverlight plug-in is displayed as a windowless plug-in. (Applies to Windows versions of Silverlight only.)

Syntax

object Element

```
<object ...>
  <param name="windowless" value="bool"/>
  ...
</object>
```

silverlight.js

```
Silverlight.CreateObject(,,,{windowless:'bool'});
-or-
Silverlight.CreateObjectEx({properties:{windowless:'bool'}});
```

JavaScript

```
            value = silverlightObject.settings.Windowless;
```

COM

IXcpControl::Settings gets an IDispatch interface, which you can use to call Windowless.

Managed Code

See Settings..::.Windowless.

ASP.NET Controls for Silverlight Client

See SilverlightWindowless()()().

Property Value

true if the Silverlight plug-in is displayed as a windowless plug-in; false if the Silverlight plug-in is displayed as a windowed plug-in. The default value is false.

Remarks

This property is settable only as an initialization parameter and is read-only for all other access models.

Important Note:

Set the Windowless property to true only when it is required. Running in windowless mode can have a significant effect on performance (for example, tearing can occur in animations). Media playback is not recommended at all in windowless mode.

When Silverlight is running under Windows, if the value is true and the Background property specifies an alpha value, the windowless mode enables HTML overlay. In this case, the HTML on the page blends with the contents of the plug-in, which includes its background.

On Macintosh computers, this property has no effect either at initialization or at run time. The behavior is always windowless.

Note:

In the Silverlight.js library, functions that reference the value of this property use the parameter named isWindowless instead of Windowless.