SVideo Control Reference

OLE Controls

An OLE control can be described as a simple “OLE object”, more specifically a “COM Object”.   This means that an OLE control must at least support the IUnknown interface, and it must be self-registering.   Controls implement only the functionality they need, allowing them to be as lightweight as possible.

SVideo Control

The Surround Video Internet Control for ActiveX(tm) provides content providers with the ability to view 360 degree panoramic images.   These images can be stored either on a local drive, or on a web server.   The control knows where to get its data by parsing its DataSourceName Property.   The control will download the data from the specified URL, using the new ActiveX(tm) technology developed by Microsoft Corporation.

A Surround Video Image consists of a number of vertical stripes that make up the complete image.   The benefit of striping the image is that any stripe can be loaded independently, so only the visible portion of the image needs to be furnished in order for rendering to occur.   Each image can have areas on them defined as Links(HyperLinks). These links can be defined as either “Internal” or “External”.   When a link is selected from within the control, the control first determines whether the link is defined as internal or external, then takes the appropriate action.   If the link is defined as internal, the control attempts to open the specified data source name for that link within itself.   If the link is defined as external, the control notifies its container that it needs to handle the opening of the specified data source name for that link.

Installation

Automatic:

From Internet Explorer, go to URL “www.bdiamond.com”, and follow the links to the Surround Video Internet Control demo page. The “Surround.dll” file and “Svideo.ocx” file will automatically be downloaded to the client machine and placed in the system directory. Both DLL’s will be automatically registered. The user will have to agree to the components certificate of authenticity when prompted by Internet Explorer. This informs the user that the components being downloaded are virus free, and will not harm the client machine.

Glossary

COM	"Component Object Model", is all the basic protocols for object communication. It defines the rules for things such as what constitutes a "COM object", and how that COM object gets instantiated.
OLE Automation	The ability of an application to control another application's objects programmatically. An integral part of OLE Automation is the ability of an object to describe its capabilities through type descriptions. This feature is absolutely key to OLE controls.
OLE Documents	All the document-centric features of OLE, such as linking and embedding, drag-and-drop, and visual editing.
OLE Controls	OLE objects that support the in-place activation features of OLE Documents in a way known as "inside out". These objects make use of OLE Automation to expose properties and methods and to provide event support.
OLE Messaging	A interface for messaging using OLE Automation.
IDispatch	The way in which object expose their properties and methods to the outside world.
Dispatch ID	The numeric identifier of a property or method.
Early Binding	Occurs when an automation controller determines whether a property or method is available at compile-time.
Late Binding	Occurs when an automation controller determines whether a property or method is available at runtime.
Container	An application that allows objects to be embedded within it, such as Microsoft Word.

ReadyState Property

Description

Indicates the current state of the control.

Syntax(Visual Basic)

[form.] object**.ReadyState** [= { **READYSTATE_UNINITIALIZED** |
READYSTATE_LOADING |
READYSTATE_LOADED |
READYSTATE_INTERACTIVE |
READYSTATE_COMPLETE }]

The syntax for the **ReadyState** property has these parts:

form An object expression that evaluates to a Visual Basic form.

object An object expression that evaluates to an **SVideo** control.

Settings

The settings for **ReadyState** are:

Type	Setting	Description
long	**READYSTATE_UNINITIALIZED / 0**	Never used except as default initialization state.
long	**READYSTATE_LOADING / 1**	Control is currently loading its properties.
long	**READYSTATE_LOADED / 2**	Control has been initialized via IPersist*::Load.
long	**READYSTATE_INTERACTIVE / 3**	Control is interactive but not all data is available.
long	**READYSTATE_COMPLETE / 4**	Control has all its data.

Remarks

This property can be used to test where in the downloading process the control is.

Design-time access: read/write

Run-time access: read/write

DataSourceName Property

Description

Name of the currently opened Surround Video Image file, or of the file the user wishes to be opened.

Syntax(Visual Basic)

[form.] object**.DataSourceName** [= BSTR]

The syntax for the **DataSourceName** property has these parts:

form An object expression that evaluates to a Visual Basic form.
object An object expression that evaluates to an **SVideo** control.
BSTR A BSTR that holds the name of the Surround Video Image to open control.

Remarks

This property is used to set the name of the image the user wishes to view.
Design-time access: read/write
Run-time access: read/write

MouseManipulation Property

Description

Indicates whether or not the mouse is enabled when over the control.

Syntax(Visual Basic)

[form.] object**. MouseManipulation**[= VARIANT]

The syntax for the **MouseManipulation** property has these parts:

form	An object expression that evaluates to a Visual Basic form.
object	An object expression that evaluates to an **SVideo** control.
VARIANT	A variant datatype that holds the value of **MouseManipulation** in its *iVal* variable.

Settings

The settings for **MouseManipulation** are:

Setting	**Description**
TRUE	Mouse is enabled.
FALSE	Mouse is disabled.

Remarks

This property is used to allow a content provider to turn mouse manipulation on or off.

Design-time access: read/write

Run-time access: read/write

Location Property

Description

Specifies the default starting location in ARC_SECONDS for the image.

Syntax(Visual Basic)

[form.] object**.Location** [= long]

The syntax for the **Location** property has these parts:

form	An object expression that evaluates to a Visual Basic form.
object	An object expression that evaluates to an **SVideo** control.
VARIANT	A variant datatype that holds the value of **Location** in its *lVal* variable.

Settings

The settings for **Location** are:

Setting	**Description**
0 to **1296000**	valid ARC_SECONDS.

Remarks

Design-time access: read/write
Run-time access: read

RotationFactors Property

Description

Indicates the speed and direction that the Surround Video Image will rotate.

Syntax(Visual Basic)

[form.] object**.RotationFactors** [= VARIANT]

The syntax for the **RotationFactors** property has these parts:

form	An object expression that evaluates to a Visual Basic form.
object	An object expression that evaluates to an **SVideo** control.
VARIANT	A variant datatype that holds the value of **Location** in its *lVal* variable.

Settings

The settings for **RotationFactors** are:

Setting	**Description**
-# to **#**	The numerical value supplied can be either a negative or positive value. The negative / positive part of the value indicates direction. Negative for right, positive for Left. The numerical part of the value indicates the number of seconds it should take for the image to make one complete rotation. If the value supplied is zero, this tells the control not to rotate at all.

Remarks

Design-time access: read/write

Run-time access: read/write

ZoomKeySet Property

Description

Specifies the zoom keys for zooming in or out.

Syntax(Visual Basic)

[form.] object**.ZoomKeySet** [= variant]

The syntax for the **ZoomKeySet** property has these parts:

form	An object expression that evaluates to a Visual Basic form.
object	An object expression that evaluates to an **SVideo** control.
VARIANT	A variant datatype that holds the value of **ZoomKeySet** in its *iVal* variable.

Settings

The settings for **ZoomKeySet** are:

Setting	**Description**
0	Sets the zoom key set to be the plus(“+”) and minus(“-”) keys.
1	Sets the zoom key set to be the letter “I” and the letter “O” keys.
2	Sets the zoom key set to be the “CNTL” and “SHIFT” keys.

Remarks

Design-time access: read/write
Run-time access: read/write

EnableHyperLinks Property

Description

Indicates whether or not HyperLinks on the image are enabled.

Syntax(Visual Basic)

[form.] object**.EnableHyperLinks** [= variant]

The syntax for the **EnableHyperLinks** property has these parts:

form An object expression that evaluates to a Visual Basic form.

object An object expression that evaluates to an **SVideo** control.

VARIANT A variant datatype that holds the value of **EnableHyperLinks** in its *iVal* variable.

Settings

The settings for **EnableHyperLinks** are:

Setting	Description
TRUE	HyperLinks are enabled.
FALSE	HyperLinks are disabled.

Remarks

Design-time access: read/write

Run-time access: read/write

AboutBox Method

Description

Display the About Box for the SVideo control.

Syntax(Visual Basic)

object**.AboutBox**

Parameters

object   An object expression that evaluates to an **SVideo** control.

Remarks

Puts up the About Box for the SVideo control, displaying copyright information.

OnReadyStateChange Event

Description

Informs the container that the ReadyState Property has changed.

Syntax(Visual Basic)

Private Sub object**_OnReadyStateChange(ByVal newState)**

Parameters

object An object expression that evaluates to an **SVideo** control.

newState The new value of the ReadyState Property.

Remarks

This event is fired to the controls container when the ReadyState Property is changed. The value gets changed at different times during a download, depending on how much of the image has actually been downloaded.

See Also

ReadyState Property

OnProgress Event

Description

Informs the container that more data has arrived during a download.

Syntax(Visual Basic)

Private Sub object**_OnProgress(ByVal percentDone)**

Parameters

object	An object expression that evaluates to an **SVideo** control.
percentDone	The percentage of data that has actually been downloaded.

Remarks

This event is fired to the controls container as data from the download arrives.

ARC_SECONDS

ARC_SECONDS is a structure consisting of two longs. The first being "latitude" the second "longitude".

Surround Video Image

A 360 degree Panoramic image.

HyperLinks

HyperLinks (as they pertain the this control) are areas within the image that the user can click on to take them to another image and or URL site.

Property Sheet

A *property sheet* is a window that allows the user to view and edit the properties of an item.

Self-Registering

OLE controls must support self-registering by implementing the DllRegisterServer, and DllUnregisterServer funtions.

ActiveX

ActiveX controls, formerly known as OLE controls or OCX controls, are required to implement the IUnknown and IClassFactory interfaces. They are components that can be inserted into Web pages or other application to reuse packaged functionality someone else programmed.

OBJECT Tag

```
<OBJECT <OBJECT>
    CLASSID="clsid:7142BA01-8BDF-11CF-9E23-0000E8A37440"
        ID=SVCntrl HEIGHT=150 WIDTH=500 HSPACE=10 ALIGN=LEFT>
    <PARAM NAME="DataSourceName" VALUE="External.svh">
    <PARAM NAME="MouseManipulation" VALUE="1">
    <PARAM NAME="Location" VALUE="0">
    <PARAM NAME="RotationTime" VALUE="5">
    <PARAM NAME="AutoScrollDirection" VALUE="1">
    <PARAM NAME="ZoomKeySet" VALUE="0">
</OBJECT>
```

OLE Controls

OLE Controls are a unique architectural oppertunity that offers a single, widely available machanism for developers to expose applications to enhancement by others and to customize the applications of others.

IUnknown Interface

The most fundamental of interfaces is called IUnknown. IUnknown contains three pointers to functions: AddRef, Release, and QueryInterface. Every other COM interface must contain the IUnknown methods.