

Introduction

This document describes WebBrowser, an ActiveX™ control that developers can use to add Internet browsing capabilities to their applications, and InternetExplorer, an OLE Automation object that developers can use to control the Microsoft® Internet Explorer (IE) application from within an application.

Using the WebBrowser Control

The WebBrowser control adds browsing, document viewing, and data downloading capabilities to your applications. It allows the user to browse sites on the Internet's World Wide Web, as well as folders in the local file system and on a network. The WebBrowser control supports Web browsing through both point-and-click hyperlinking and Uniform Resource Locator (URL) navigation. Also, the control maintains a history list that allows the user to browse forward and backward through previously browsed sites, folders, and documents.

The WebBrowser control includes support for parsing and displaying HTML-encoded documents. It is also an ActiveX document container that can host any ActiveX document. This means that richly formatted documents, such as a Microsoft Excel spreadsheet or Microsoft Word documents, can be opened and edited in-place from within the WebBrowser control. WebBrowser is also an ActiveX control container that can host any ActiveX control.

Possible Uses

Some possible uses of the WebBrowser control are:

- To enable the user to navigate to and view HTML documents on the local computer, network, or World Wide Web.
- To provide a single frame in which the user can view and edit all types of ActiveX documents.
- To create a customized Web browsing application based on the WebBrowser control.

Navigate to a New Location with the Navigate Method

The WebBrowser control can browse to any location in the local file system, on the network, or on the World Wide Web. You use the Navigate method to tell the control which location to browse to. The first parameter is a string that contains the name of the location. To browse to a location in the local file system or on the network, specify the full path to the file system location or the Universal Naming Convention (UNC) name of the location on the network. To browse to a site on the World Wide Web, specify the URL of the site. By including a text box in your application, you can let the user specify the location to browse to, and then pass the location to the Navigate method.

The Navigate method allows you to target a specific frame on an HTML page, causing the WebBrowser control to display a Web site or file system location in that frame. First, you would call the Navigate method and specify the URL of an HTML page that contains a frame. Then, by specifying the name of the frame in subsequent calls to Navigate, you can direct the control to display subsequent locations within that frame.

You can use the LocationName and LocationURL properties to retrieve information about the location that the WebBrowser control is currently displaying. If the location is an HTML page on the World Wide Web, LocationName retrieves the title of that page, and LocationURL retrieves the URL of that page. If the location is a folder or file on the network or local computer, LocationName and LocationURL both retrieve the UNC or full path of the folder or file.

An HTML author can create a hyperlink that causes an Internet server to execute a program whenever the hyperlink is selected. The hyperlink typically passes data to the program during the HTTP transaction, and the program processes the data. The Navigate method supports navigating to such a hyperlink, and includes parameters that allow you to specify data to pass to the program. You can specify an HTTP header, HTTP post data, and the URL of the referring document. The HTTP header specifies such things as the action required of the server, the type of data being passed to the server, or a status code. The HTTP post data is the actual data for the program to process. The referring document is the HTML document that contains the hyperlink.

Update Cached Pages with the Refresh or Refresh2 Method

The WebBrowser control stores Web pages from recently visited sites in cached memory on the user's hard disk so that, when the control revisits a site, it can save time by reloading the page from the cache rather than downloading it again from the Internet server. You can force the WebBrowser control to re-download a page by using the [Refresh](#) or [Refresh2](#) method—this ensures that the user is viewing the current version of the page. Also, you can prevent the control from using the cache by specifying the [navNoReadFromCache](#) and [navNoWriteToCache](#) flags when calling the [Navigate](#) method.

Navigate the History List with the GoBack and GoForward Methods

During a browsing session, the WebBrowser control maintains a history list of all Web sites that it visited during the session (unless you specify navNoHistory when calling the Navigate method). You can direct the control to browse backward and forward through the sites in the list by using the GoBack and GoForward methods.

You can also use the GoHome method to cause the WebBrowser control to return to the user's home or start page, and the GoSearch method to go to the user's Web searching page.

Control Browsing Operations with the Busy Property and Stop Method

You can use the Busy property to determine if the WebBrowser control is in the process of navigating to a new location or downloading a file. If the control is busy, you can use the Stop method to cancel the navigation or the download before it is finished.

Hide or Show the WebBrowser Control with the Visible Property

By default, the WebBrowser control is hidden when it is first created, but it becomes visible after calling the [Navigate](#) or [GoSearch](#) method. You can also set the [Visible](#) property to **True** to show the control, or to **False** to hide it.

WebBrowser Events

The WebBrowser control fires a number of different events to notify an application of user- and browser-generated activity. When the control is about to navigate to a new location, it fires a BeforeNavigate event that specifies the URL or path of the new location and any other data that will be transmitted to the Internet server through the HTTP transaction. The data can include the HTTP header, HTTP post data, and the URL of the referrer. The BeforeNavigate event also includes a cancel flag that you can set to **False** to cancel the navigation. The WebBrowser control fires the NavigateComplete event after it has navigated to a new location. This event includes the same information as BeforeNavigate, except NavigateComplete does not include the cancel flag.

Whenever the WebBrowser control is about to begin a download operation, it fires the DownloadBegin event. The control fires a number of ProgressChange events as the operation progresses, and then fires the DownloadComplete event after completing the operation. Applications typically use these three events to indicate the progress of the download operation, often by displaying a progress bar. An application would show the progress bar in response to DownloadBegin, update the progress bar in response to ProgressChange, and hide it in response to DownloadComplete.

When an application calls the Navigate method with the navOpenInNewWindow flag, the WebBrowser control fires the NewWindow event before navigating to the new location. The event includes information about the new location, and a flag that indicates whether the application or the control is to create the new window. Set this flag to **True** if your application will create the window, or to **False** if the WebBrowser control should create it.

Using the InternetExplorer Object

The InternetExplorer object allows an application to create and manipulate an instance of Internet Explorer through OLE Automation. The InternetExplorer object supports the same properties and methods as the WebBrowser control, plus several that the WebBrowser control does not support. This section describes the additional properties and methods supported by the InternetExplorer object. For information about the properties and methods supported by both objects, see [Using the WebBrowser Control](#).

Show and Hide User Interface Elements

The `InternetExplorer` object supports a number of properties that you can use to show and hide user-interface elements of Internet Explorer, including the menu bar, the status bar, and the toolbar. The `MenuBar`, `StatusBar`, and `ToolBar` properties all take Boolean values. If the value of any of these properties is **True**, the corresponding user interface element is shown; if the value is **False**, the corresponding element is hidden. You can set or retrieve the text of the status bar by using the `StatusText` property.

The `FullScreen` property determines whether Internet Explorer is in full-screen or normal window mode. In full-screen mode, the Internet Explorer main window is maximized and the status bar, toolbar, menu bar, and title bar are hidden.

Retrieve Information About Internet Explorer

You can use the Name property to retrieve the name of the Internet Explorer application, or the Path property to retrieve the application's full path to the Internet Explorer application. The FullName property is similar to Path, except it retrieves the full path, including the file name, of the executable file that contains the Internet Explorer application. The HWND property retrieves the window handle of the Internet Explorer main window.

Use the Quit method to close the Internet Explorer application.

WebBrowser Object

The **WebBrowser** object is an ActiveX control that allows you to add browsing capabilities to your applications. The WebBrowser control can be used to browse sites on the World Wide Web, as well as directories on the local computer and on network servers.

Properties

Application, Busy, Container, Document, Height, Left, LocationName, LocationURL, Parent, Top, TopLevelContainer, Type, Width

Methods

GoBack, GoForward, GoHome, GoSearch, Navigate, Refresh, Refresh2, Stop

Events

BeforeNavigate, CommandStateChange, DownloadBegin, DownloadComplete, FrameBeforeNavigate, FrameNavigateComplete, FrameNewWindow, NavigateComplete, NewWindow, ProgressChange, StatusTextChange, TitleChange

InternetExplorer Object

The InternetExplorer object allows an application to create and control an instance of the Microsoft Internet Explorer application.

Properties

Application, Busy, Container, Document, FullName, FullScreen, Height, HWND, Left, LocationName, LocationURL, MenuBar, Name, Parent, Path, StatusBar, StatusText, ToolBar, Top, TopLevelContainer, Type, Visible, Width

Methods

ClientToWindow, GetProperty, GoBack, GoForward, GoHome, GoSearch, Navigate, PutProperty, Quit, Refresh, Refresh2, Stop

Events

BeforeNavigate, CommandStateChange, DownloadBegin, DownloadComplete, FrameBeforeNavigate, FrameNavigateComplete, FrameNewWindow, NavigateComplete, NewWindow, ProgressChange, PropertyChange, Quit, StatusTextChange, TitleChange

Application Property

object.Application

Returns the automation object supported by the application that contains the WebBrowser control if the object is accessible; otherwise, this property returns the WebBrowser control's automation object.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

Busy Property

object.Busy

Returns a Boolean value specifying whether the WebBrowser control or Internet Explorer is engaged in a navigation or downloading operation.

- The **Busy** property returns these values:

Value	Description
True	A download or other operation is in progress.
False	No download or other operation is in progress.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

Container Property

object.Container

Returns an object that evaluates to the container of the WebBrowser control, if any.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

Document Property

object.Document

Returns the automation object of the active document, if any.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

FullName Property

object.FullName

Returns a string that evaluates to the fully qualified path of the executable file that contains the Internet Explorer application.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

InternetExplorer

FullScreen Property

object.**FullScreen** [= *value*]

Returns or sets a value indicating whether Internet Explorer is in full-screen or normal window mode. In full-screen mode, the Internet Explorer main window is maximized and the status bar, toolbar, menu bar, and title bar are hidden.

object

Required. An object expression that evaluates to an object in the Applies To list.

value

Optional. A Boolean expression that determines whether Internet Explorer is in full-screen or normal window mode. If **True**, the object is in full-screen mode; if **False**, it is in normal mode.

Applies To

InternetExplorer

Height Property

object.Height [= *height*]

Returns or sets the vertical dimension, in pixels, of the frame window that contains the WebBrowser control.

object

Required. An object expression that evaluates to an object in the Applies To list.

height

Optional. A long integer value specifying the vertical dimension of the frame window, in pixels.

Applies To

WebBrowser, InternetExplorer

HWND Property

object.HWND

Returns the handle of the Internet Explorer main window.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

InternetExplorer

Left Property

object.Left [= *distance*]

Returns or sets the distance between the internal left edge of the WebBrowser control and the left edge of its container.

object

Required. An object expression that evaluates to an object in the Applies To list.

distance

Optional. A long integer expression specifying the distance between the internal left edge of the WebBrowser control and the left edge of its container.

The **Left** property is measured in units depending on the coordinate system of its container. The values for this property change as the object is moved by the user or by code.

Applies To

WebBrowser, InternetExplorer

LocationName Property

object.LocationName

Returns a string that contains the name of the resource that the WebBrowser control is currently displaying. If the resource is an HTML page on the World Wide Web, the name is the title of that page. If the resource is a folder or file on the network or local computer, the name is the UNC or full path of the folder or file.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

LocationURL Property

object.LocationURL

Returns a string that contains the URL of the resource that the WebBrowser control or Internet Explorer is currently displaying. If the resource is a folder or file on the network or local computer, the name is the UNC or full path of the folder or file.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

MenuBar Property

object.MenuBar [= *value*]

Returns or sets a value that determines whether the Internet Explorer menu bar is visible or hidden.

object

Required. An object expression that evaluates to an object in the Applies To list.

value

Optional. A Boolean expression that determines whether the menu bar is visible. If **True**, the menu bar is visible; if **False**, it is hidden.

Applies To

InternetExplorer

Name Property

object.Name

Returns a string that evaluates to the name of the Internet Explorer application; that is, "Microsoft Internet Explorer."

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

InternetExplorer

Parent Property

object.Parent

Returns the form on which the WebBrowser control is located, or the automation object supported by Internet Explorer.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

Path Property

object.Path

Returns a string that evaluates to the full path of the Internet Explorer application.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

InternetExplorer

StatusBar Property

object.StatusBar [= *value*]

Returns or sets a value that determines whether the status bar is visible.

object

Required. An object expression that evaluates to an InternetExplorer object.

value

Optional. A Boolean expression that determines whether the status bar is visible. If **True**, the status bar is visible; if **False**, it is hidden.

Applies To

InternetExplorer

StatusText Property

object.StatusText [= *value*]

Returns or sets the text for the status bar.

object

Required. An object expression that evaluates to an InternetExplorer object.

value

Optional. A string that evaluates to the text for the status bar.

Applies To

InternetExplorer

ToolBar Property

object.ToolBar [= *value*]

Returns or sets a value that determines whether the toolbar is visible.

object

Required. An object expression that evaluates to an InternetExplorer object.

value

Optional. A Boolean expression that determines whether the toolbar is visible. If **True**, the toolbar is visible; if **False**, it is hidden.

Applies To

InternetExplorer

Top Property

object.**Top** [= *value*]

Returns or sets the distance between the internal top edge of the WebBrowser control and the top edge of its container.

object

Required. An object expression that evaluates to an object in the Applies To list.

value

Optional. A long integer expression specifying distance.

The **Top** property is measured in units depending on the coordinate system of its container. The values for this property change as the object is moved by the user or by code.

Applies To

WebBrowser, InternetExplorer

TopLevelContainer Property

object.TopLevelContainer

Returns a Boolean value indicating whether the given object is a top-level container.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

Type Property

object.Type

Returns a string expression that specifies the type name of the contained document object.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

Visible Property

object.Visible [= *value*]

Returns or sets a value indicating whether Internet Explorer is visible or hidden.

object

Required. An object expression that evaluates to an object in the Applies To list.

value

Optional. A Boolean expression specifying the visible state of Internet Explorer. If **True**, the window is visible; if **False**, it is hidden.

Applies To

InternetExplorer

Width Property

object.Width [= *width*]

Returns or sets the horizontal dimension, in pixels, of the frame window that contains the WebBrowser control.

object

Required. An object expression that evaluates to an object in the Applies To list.

width

Optional. A long integer value specifying the horizontal dimension of the frame window, in pixels.

Applies To

WebBrowser, InternetExplorer

ClientToWindow Method

object.**ClientToWindow** *pcx*, *pcy*

Converts the client coordinates of a point to window coordinates. Client coordinates are relative to the upper-left corner of the client area; window coordinates are relative to the upper-left corner of a window.

object

Required. An object expression that evaluates to an object in the Applies To list.

pcx

Required. A long integer value that specifies the x-coordinate of the point in client coordinates. When **ClientToWindow** returns, this variable contains the x-coordinate of the point in window coordinates.

pcy

Required. A long integer value that specifies the y-coordinate of the point in client coordinates. When **ClientToWindow** returns, this variable contains the y-coordinate of the point in window coordinates.

Applies To

InternetExplorer

GetProperty Method

object.GetProperty szProperty, vtValue

Retrieves the current value of a property associated with the given object.

object

Required. An object expression that evaluates to an object in the Applies To list.

szProperty

Required. A string expression that contains the name of the property to retrieve.

vtValue

Required. A variable that receives the current value of the property.

Applies To

[InternetExplorer](#)

GoBack Method

object.GoBack

Navigates backward one item in the history list.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

GoForward Method

object.GoForward

Navigates forward one item in the history list.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

GoHome Method

object.GoHome

Navigates to the current home or start page, as specified in the Internet Explorer Options dialog box and Internet Control Panel.

object

Required. An object expression that evaluates to object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

GoSearch Method

object.GoSearch

Navigates to the current search page, as specified in the Internet Explorer Options dialog box and Internet Control Panel.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

Navigate Method

object.Navigate URL [*Flags*,] [*TargetFrameName*,] [*PostData*,] [*Headers*]

Navigates to the resource identified by a Universal Resource Locator (URL), or to the file identified by a full path.

object

Required. An object expression that evaluates to an object in the Applies To list.

URL

Required. A string expression that evaluates to the URL of the resource to display, or the full path of the file to display.

Flags

Optional. A constant or value that specifies whether to add the resource to the history list, whether to read from or write to the cache, and whether to display the resource in a new window. It can be a combination of the following values.

Constant	Value	Meaning
navOpenInNewWindow	1	Open the resource or file in a new window.
navNoHistory	2	Do not add the resource or file to the history list. The new page replaces the current page in the list.
navNoReadFromCache	4	Do not read from the disk cache for this navigation.
navNoWriteToCache	8	Do not write the results of this navigation to the disk cache.

TargetFrameName

Optional. A string expression that evaluates to the name of a frame in which to display the resource.

PostData

Optional. Data to send to the server during the HTTP POST transaction. For example, the POST transaction is used to send data gathered by an HTML form. If this parameter does not specify any post data, the **Navigate** method issues an HTTP GET transaction. This parameter is ignored if *URL* is not an HTTP URL.

Headers

Optional. A value that specifies additional HTTP headers to send to the server. These headers are added to the default Internet Explorer headers. The headers can specify such things as the action required of the server, the type of data being passed to the server, or a status code. This parameter is ignored if *URL* is not an HTTP URL.

Applies To

WebBrowser, InternetExplorer

PutProperty Method

object.PutProperty szProperty, vtValue

Sets the value of a property associated with the given object.

object

Required. An object expression that evaluates to an object in the Applies To list.

szProperty

Required. A string expression that contains the name of the property to set.

vtValue

Required. A variable that specifies the new value of the property.

Applies To

InternetExplorer

Quit Method

object.Quit

Closes the Internet Explorer application.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

InternetExplorer

Refresh Method

object.Refresh

Reloads the page that the WebBrowser control is currently displaying.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

Refresh2 Method

object.Refresh2 [*Level*]

Reloads the file that the WebBrowser control is currently displaying. Unlike the [Refresh](#) method, this method contains a parameter that specifies the refresh level.

object

Required. An object expression that evaluates to an object in the Applies To list.

Level

Optional. A constant or value that specifies the refresh level. It can be one of the following constants or values.

Constant	Value	Meaning
REFRESH_NORMAL	0	Perform a lightweight refresh that does not include sending the HTTP "pragma:nocache" header to the server.
REFRESH_IFEXPIRED	1	Perform a lightweight refresh if the page has expired.
REFRESH_COMPLETELY	3	Perform a full refresh that includes sending a "pragma:nocache" header to the server (HTTP URLs only).

The "pragma:nocache" header tells the server not to return a cached copy, but to ensure that the information is as fresh as possible. Browsers typically send this header when the user selects refresh, but the header causes problems for some servers.

Applies To

[WebBrowser](#), [InternetExplorer](#)

Stop Method

object.**Stop**

Cancels any pending navigation or download operation, and stops any dynamic page elements such as background sounds and animations.

object

Required. An object expression that evaluates to an object in the Applies To list.

Applies To

WebBrowser, InternetExplorer

BeforeNavigate Event

Private Sub *object*_BeforeNavigate(ByVal URL As String, ByVal Flags As Long, ByVal TargetFrameName As String, PostData As Variant, ByVal Headers As String, Cancel As Boolean)

Occurs when the WebBrowser control is about to navigate to a different URL, which may happen as a result of external automation, internal automation from a script, or the user clicking a link or typing in the address bar. The container has an opportunity to cancel the pending navigation.

object

Required. An object expression that evaluates to an object in the Applies To list.

URL

A string expression that evaluates to the URL to which the browser is navigating.

Flags

Reserved for future use.

TargetFrameName

A string expression that evaluates to the name of the frame in which to display the resource, or NULL if no named frame is targeted for the resource.

PostData

Data to send to the server if the HTTP POST transaction is being used.

Headers

A value that specifies the additional HTTP headers to send to the server (HTTP URLs only). The headers can specify such things as the action required of the server, the type of data being passed to the server, or a status code.

Cancel

A Boolean value that the container can set to **True** to cancel the navigation operation, or to **False** to allow it to proceed.

Applies To

WebBrowser, InternetExplorer

CommandStateChange Event

Private Sub *object_CommandStateChange* (**ByVal** *Command* **As Long**,
ByVal *Enable* **As Boolean**)

Occurs when the enabled state of a command changes.

object

An object expression that evaluates to an object in the Applies To list.

Command

A long integer specifying the identifier of the command that changed. It can be one of the following constants or values.

Constant	Value	Meaning
CSC_UPDATECOMMANDS	-1	The enabled state of a toolbar button may have changed; the <i>Enable</i> parameter should be ignored.
CSC_NAVIGATEFORWARD	1	The enabled state of the Forward button has changed.
CSC_NAVIGATEBACK	3	The enabled state of the Back button has changed.

Enable

A Boolean value that is **True** if the command is enabled, or **False** if not.

Applies To

WebBrowser, InternetExplorer

DownloadBegin Event

Private Sub *object*_DownloadBegin ()

Occurs when a navigation operation is beginning. This event is fired shortly after the [BeforeNavigate](#) event, unless the navigation is canceled. Any animation or "busy" indication that the container needs to display should be connected to this event.

object

An object expression that evaluates to an object in the Applies To list.

Applies To

[WebBrowser](#), [InternetExplorer](#)

DownloadComplete Event

Private Sub *object*_DownloadComplete ()

Occurs when a navigation operation finished, was halted, or failed. Unlike [NavigateComplete](#), which is fired only when a URL is successfully navigated to, this event is always fired after a navigation starts. Any animation or "busy" indication that the container needs to display should be connected to this event.

object

An object expression that evaluates to an object in the Applies To list.

Applies To

[WebBrowser](#), [InternetExplorer](#)

FrameBeforeNavigate Event

Private Sub *object*_FrameBeforeNavigate(**ByVal** *URL* As String, **ByVal** *Flags* As Long, **ByVal** *TargetFrameName* As String, *PostData* As Variant, **ByVal** *Headers* As String, *Cancel* As Boolean)

Occurs when the WebBrowser control is about to navigate to a different URL, which may happen as a result of external automation, internal automation from a script, or the user clicking a link or typing in the address bar. The container has an opportunity to cancel the pending navigation. This event is fired for navigations that occur inside an HTML frame.

object

Required. An object expression that evaluates to an object in the Applies To list.

URL

A string expression that evaluates to the URL to which the browser is navigating.

Flags

Reserved for future use.

TargetFrameName

A string expression that evaluates to the name of the frame in which to display the resource, or NULL if no named frame is targeted for the resource.

PostData

Data to send to the server if the HTTP POST transaction is being used.

Headers

A value that specifies the additional HTTP headers to send to the server (HTTP URLs only). The headers can specify such things as the action required of the server, the type of data being passed to the server, or a status code.

Cancel

A Boolean value that the container can set to **True** to cancel the navigation operation, or to **False** to allow it to proceed.

Applies To

WebBrowser, InternetExplorer

FrameNavigateComplete Event

Private Sub *object*_FrameNavigateComplete(ByVal *URL* As String)

Occurs after the browser has successfully navigated to a new location. This event is fired for navigations that occur inside an HTML frame. The document may still be downloading (and in the case of HTML, images may still be downloading), but at least part of the document has been received from the server, and the viewer for the document has been created.

object

An object expression that evaluates to an object in the Applies To list.

URL

A string expression that evaluates to the URL that was navigated to. Note that this URL can be different from the URL that the browser was told to navigate to. One reason is that this URL is the canonicalized and qualified URL; for instance, if an application specified a URL of "www.microsoft.com" in a call to the [Navigate](#) method, the URL passed by [NavigateComplete](#) will be "http://www.microsoft.com/". Also, if the server has redirected the browser to a different URL, the redirected URL will be reflected here.

Applies To

[WebBrowser](#), [InternetExplorer](#)

FrameNewWindow Event

Private *object* **FrameNewWindow** (ByVal *URL* As String, ByVal *Flags* As Long, ByVal *TargetFrameName* As String, *PostData* As Variant, ByVal *Headers* As String, *Processed* As Boolean)

Occurs when a new window is to be created for displaying a resource. This event is fired for navigations that occur inside an HTML frame. Some actions that can cause this include the user shift-clicking on a link, the user right-clicking on a link and choosing "open in new window", or a targeted navigation to a frame name that does not yet exist. The container has an opportunity to handle the new window creation itself. If it does not, a top-level Internet Explorer window is created as a separate process.

object

An object expression that evaluates to an object in the Applies To list.

URL

A string expression that evaluates to the URL of the resource being navigated to.

Flags

Reserved for future use.

TargetFrameName

A string expression that evaluates to the name of the frame in which to display the resource, or NULL if no named frame is targeted for the resource.

PostData

Data to send to the server if an HTTP POST transaction is used.

Headers

A value that specifies the HTTP headers to send to the server (HTTP URLs only). The headers can specify such things as the action required of the server, the type of data being passed to the server, or a status code.

Processed

A Boolean value that indicates whether the container intends to create the new window. Set this parameter to **True** if the container will create the window, or to **False** if a top-level Internet Explorer window is to be created.

The preferred behavior of WebBrowser control containers is to process this event, create a new instance of the WebBrowser control, and pass all the parameters from the [NewWindow](#) event directly to the [Navigate](#) method on the newly created WebBrowser control. Another option for containers that cannot or do not need to create a new window is to degrade by performing the navigation in the existing window. To do this, they may process this event and then pass the parameters from this event to [Navigate](#) on the existing window.

Applies To

[WebBrowser](#), [InternetExplorer](#)

NavigateComplete Event

Private Sub *object*_NavigateComplete(ByVal *URL* As String)

Occurs after the browser has successfully navigated to a new location. The document may still be downloading (and in the case of HTML, images may still be downloading), but at least part of the document has been received from the server, and the viewer for the document has been created.

object

An object expression that evaluates to an object in the Applies To list.

URL

A string expression that evaluates to the URL that was navigated to. Note that this URL can be different from the URL that the browser was told to navigate to. One reason is that this URL is the canonicalized and qualified URL; for instance, if an application specified a URL of "www.microsoft.com" in a call to the [Navigate](#) method, the URL passed by **NavigateComplete** will be "http://www.microsoft.com/". Also, if the server has redirected the browser to a different URL, the redirected URL will be reflected here.

Applies To

WebBrowser, InternetExplorer

NewWindow Event

Private *object_NewWindow* (**ByVal** *URL As String*, **ByVal** *Flags As Long*,
ByVal *TargetFrameName As String*, *PostData As Variant*, **ByVal** *Headers As String*,
Processed As Boolean)

Occurs when a new window is to be created for displaying a resource. Some actions that can cause this include the user shift-clicking on a link, the user right-clicking on a link and choosing "open in new window", or a targeted navigation to a frame name that does not yet exist. The container has an opportunity to handle the new window creation itself. If it does not, a top-level Internet Explorer window is created as a separate process.

object

An object expression that evaluates to an object in the Applies To list.

URL

A string expression that evaluates to the URL of the resource being navigated to.

Flags

Reserved for future use.

TargetFrameName

A string expression that evaluates to the name of the frame in which to display the resource, or NULL if no named frame is targeted for the resource.

PostData

Data to send to the server if an HTTP POST transaction is used.

Headers

A value that specifies the HTTP headers to send to the server (HTTP URLs only). The headers can specify such things as the action required of the server, the type of data being passed to the server, or a status code.

Processed

A Boolean value that indicates whether the container intends to create the new window. Set this parameter to **True** if the container will create the window, or to **False** if a top-level Internet Explorer window is to be created.

The preferred behavior of WebBrowser control containers is to process this event, create a new instance of the WebBrowser control, and pass all the parameters from the **NewWindow** event directly to the Navigate method on the newly created WebBrowser control. Another option for containers that cannot or do not need to create a new window is to degrade by performing the navigation in the existing window. To do this, they may process this event and then pass the parameters from this event to Navigate on the existing window.

Applies To

WebBrowser, InternetExplorer

ProgressChange Event

Private Sub *object*_ProgressChange(ByVal *Progress* As Long, ByVal *ProgressMax* As Long)

Occurs when the progress of a download operation is updated.

object

An object expression that evaluates to an object in the Applies To list.

Progress

A long integer that specifies the amount of total progress to show, or -1 when progress is complete.

ProgressMax

A long integer that specifies the maximum progress value.

The container can use the information provided by this event to display the number of bytes downloaded so far or to update a progress indicator.

To calculate the percentage of progress to show in a progress indicator, multiply the value of *Progress* by 100 and divide by the value of *ProgressMax* (unless progress is -1, in which case the container can indicate that the operation is finished or hide the progress indicator).

Applies To

WebBrowser, InternetExplorer

PropertyChange Event

Private Sub *object*_PropertyChange(ByVal *szProperty* As String)

Occurs when the PutProperty method changes the value of a property.

object

An object expression that evaluates to an object in the Applies To list.

szProperty

A string expression that contains the name of the property whose value has changed.

Applies To

InternetExplorer

Quit Event

Private Sub *object_Quit*(*Cancel* **As Boolean**)

Occurs when the Internet Explorer application is ready to quit.

object

An object expression that evaluates to an object in the Applies To list.

Cancel

A Boolean value that is **True** if the last Quit was canceled, or **False** if not.

Applies To

InternetExplorer

StatusTextChange Event

Private Sub *object*_StatusTextChange(ByVal *Text* As String)

Occurs when the status bar text has changed.

object

An object expression that evaluates to an object in the Applies To list.

Text

A string containing the new status bar text.

The container can use the information provided by this event to update the text of a status bar.

Applies To

WebBrowser, InternetExplorer

TitleChange Event

Private Sub *object*_TitleChange(ByVal *Text* As String)

Occurs when the title of a document in the WebBrowser control becomes available or changes. For HTML, the title may change; while HTML is still downloading, the URL of the document is set as the title. After the real title (if there is one) is parsed from the HTML, the title is changed to reflect the actual title.

object

An object expression that evaluates to an object in the Applies To list.

Text

A string containing the new document title.

Applies To

WebBrowser, InternetExplorer

