

Using the HTML Control

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmscHTMLOverviewC"}

The Hyper Text Markup Language (HTML) is used for creating World Wide Web documents. Documents created in HTML are transmitted using the Hyper Text Transfer Protocol (HTTP). The **HTML** control allows you to create applications that include an HTML viewer.

The **HTML** control can also be used as non-visual parser. Used in this mode, each element of an HTML document can be examined, categorized, and treated according to predefined rules.

For more information on the HTML language, see "Internet References."

The **HTML** control supports the following features:

- Scrollable view of selected page.
- Inline graphics: GIF, JPEG, BMP, XBM.
- HTML version 2.X plus most NetScape 2.0 and Explorer 2.0 extensions.
- Built-in document retrieval for HTTP and File URLs.
- Built-in HTTP form execution.
- Properties controlling the style sheet (such as fonts and colors).
- **DocInput**, **DocOutput** object interfaces for flexible data transfer.
- Events for overriding default processing.

Possible Uses

The **HTML** control can be used in the following scenarios:

- To create a Web browser that limits the viewer to selected web sites.
- To create a Web browser that allows the user to view only the non-graphic elements of a page.
- To integrate HTML browsing into a corporate application, allowing users to view content on the corporation's Intranet.

Scenario: Creating a Web browser

A basic Web browser can be created easily with just one line of code. The **RequestDoc** method takes one argument, the URL of a Web page. However, beyond the basics of connecting to the Web, you will also want to keep a history of the sites that have been visited. This is accomplished using a **ComboBox** control. The steps to creating this simplest of Web browser are as follows:

1. Use a **ComboBox** and the **RequestDoc** method to retrieve and view an HTML document.
2. Use the **DoRequestDoc** event to inform the user of the status
3. Use the **BeginRetrieval** event to add the URL to the **ComboBox** list
4. Use the **RequestDoc** method to retrieve a visited document
5. Use the **Error** event to inform the user of status

Setup

The following Visual Basic controls are used in the scenario:

- 1 **HTML** control
- 1 **ComboBox** control
- 1 **Label** control

Use a ComboBox Control and the RequestDoc Method to Retrieve an HTML Document

To contain the URL of the target HTML document, you can use a **ComboBox** control. The code below

shows the **KeyDown** event of the **ComboBox** control where the **RequestDoc** method is invoked to retrieve the target document. The code tests to see if the user has pressed the return key. If so, the **RequestDoc** method is invoked.

```
Sub cmbURL_KeyDown(KeyCode As Integer, _
Shift As Integer)
    ' The combo box is named "cmbURL."
    If KeyCode = 13 Then
        HTML1.RequestDoc cmbURL.Text
    End If
End Sub
```

Use the DoRequestDoc Event to Inform the User of the Status

Retrieving an HTML document is not always be successful. (For example, the server can be busy.) As a courtesy to your user, you can program a **Label** control to contain the status of the retrieval. Whenever the **HTML** control invokes the **RequestDoc** method, the **DoRequestDoc** event occurs. This event occurs even if the retrieval is not successful. The code below uses the event to inform the user of the status of the retrieval.

```
Sub HTML1_DoRequestDoc(ByVal URL as String, ByVal _
Element As HTMLElement, ByVal DocInput As DocInput _
EnableDefault As Boolean)
    ' The label control is called "lblStatus"
    lblStatus.Caption = "Contacting: " & URL
End Sub
```

Use the BeginRetrieval Event to Add the URL to the ComboBox List

When a successful retrieval is initiated, the **BeginRetrieval** event occurs. In this event, you can insert the code that maintains the state of the user interface. For example, the combo box which contains a history of the documents already retrieved can be updated here. The code below iterates through the combo box and checks if the current document is a duplicate. If the URL is found, the code exits; if it's not found, the code adds the URL to the combo box's list.

```
Sub HTML1_BeginRetrieval()
    Dim strURL As String ' String variable for the URL.
    strURL = HTML1.URL ' Set the variable to the
                        ' the URL property.
    Dim bFound As Boolean ' Flag for search.
    Dim i As Integer ' Counter variable.
    For i = 0 To cmbURL.ListCount - 1
        If cmbURL.List(i) = strURL Then
            bFound = True
            Exit For
        End If
    Next i

    If Not bFound Then
        cmbURL.AddItem strURL ' Add the URL to the
                              ' combo box.
    End If
    cmbURL.Text = strURL
End Sub
```

Use the RequestDoc Method to Retrieve a Visited Document

After the user has retrieved several documents, the **ComboBox** control contains a history of the visited documents. Of course, the history is useless unless there is an easy way to retrieve any of the

documents. The code below uses the **ComboBox** control's Click event to retrieve any document. Once again, the **RequestDoc** method is used.

```
Sub cmbURL_Click()  
    HTML1.RequestDoc cmbURL.Text  
End Sub
```

Use the Error Event to Inform the User of Status

Retrieving HTML documents can generate several kinds of errors. Servers can be busy, for example, or the document can be unavailable. In these cases, use the Error event to inform the user of the retrieval status. The Error event passes several arguments that can easily be used to update the **Label** control. The code below passes the Number and Description arguments to the **Label** control's **Caption** property.

```
Sub HTML1_Error(Number As Integer, Description As _  
String, Scode As Long, Source As String, HelpFile As _  
String, HelpContext As Long, CancelDisplay As Boolean)  
    lblStatus.Caption = Number & ": " & Description  
End Sub
```

Suggestions for Retrieving HTML Data

HTML source text or graphics data can be retrieved in the following ways:

- Explicitly. -You can invoke the **RequestDoc** and **RequestSubmit** methods to explicitly specify a new main document by URL or request submission of a form. These methods cause the DoRequestDoc and DoRequestSubmit events to occur.
- By clicking an active link and causing the DoRequestDoc event to occur. When you click on an active link, the DoRequestDoc event occurs to request retrieval of a new main document identified by the URL of the link. The default for this event is to retrieve the document using HTTP or from a local file.
- By parsing the DoRequestEmbedded event for embedded documents that are to be displayed inline. The default for this event is to retrieve the document using HTTP or from a local file.
- By parsing the DoRequestSubmit event to request form submission. When you click on a form submission button, the DoRequestSubmit event is activated to request form submission using HTTP. The response is used as the next main document.

Using the HTML Control as a Non-Visual HTML Parser

The **HTML** Control can also be used as a non-visual HTML parser. If the control is set to be invisible at run-time, no view window is created. When HTML input data is processed and the **ElemNotification** property is set to **True**, the **DoNewElement** event is activated as each element is parsed. You can then extract the attributes of the parsed element. If this event is canceled, parsing will continue but the **HTML** Control will not store the element. For more information, see "Using the **HTMLAttr** Object and **HTMLAttrs** Collection."

Using the HTMLAttr Object and HTMLAttrs Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmscUsingHTMLAttrHTMLFormObjectsC"}

The **HTMLAttr** object is used to store a single attribute of an HTML document element. A collection of attributes, the **HTMLAttrs** collection, is passed as an argument of the DoNewElement event. This event has other arguments, that, together with the **HTMLAttrs** collection, allow you to iterate through the collection members and programmatically extract valuable information about the HTML document.

Possible Uses

Using the DoNewElement event, the **HTMLAttrs** collection, and the **HTMLAttr** object, you can:

- Extract just the hyperlinks (URLs to other HTML documents) embedded in an HTML document. This collection of URLs can then be put into a database for later processing.
- Extract just the paths of inline images (usually GIF files) embedded in a document. If you are administering a World Wide Web site with GIF images that may change, this information could be stored in a database for maintenance.

Scenario: Extract the all embedded URLs from an HTML Document

To extract the URLs embedded in an HTML document, the following steps are required:

1. Set the **ElemNotification** property of the HTML control to True.
2. Invoke the **RequestDoc** method to retrieve an HTML document.
3. Determine element types in the DoNewElement event using the **ElemType** argument.
4. Extract attributes using the **For Each** statement with each member of the collection.
5. Write the data to a text file using the **Open** and **Print** statements.

Set the ElemNotification Property to True

To use the DoNewElement event, you must first set the **ElemNotification** property to **True**. The default of this property is **False**. Thus, either when initializing the control, or before retrieving an HTML document, you must have the following code:

```
HTML1.ElemNotification = True
```

Using the DoNewElement event is off by default since parsing a document can slow rendering of the HTML page. If you have no need to render the page, set the **Visible** property of the control to **False** to speed execution of the code.

Use the RequestDoc Method to Retrieve an HTML Document

To retrieve a document, you must invoke the **RequestDoc** method. This method requires one argument, the URL of the document to be retrieved. The code below assumes you have a **TextBox** control named "txtURL" where the user can type in a URL.

```
Sub txtURL_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = 13 Then ' User pressed return.
        HTML1.RequestDoc txtURL.Text
    End If
End Sub
```

Consequently, as the document is retrieved, the DoNewElement event occurs as each element is passed to the control.

Determine Element Types in the DoNewElement Event Using the ElemType Argument

The **HTML** control is capable of parsing every document into its component elements through the DoNewElement event, which passes the argument *ElemType*. The *ElemType* argument contains the

element type of the element. and you can use this argument to determine what to do with the current element. In the present example, we want to extract only URLs that are embedded in the document. The element type that contains URLs is denoted by "a."

Thus the code below tests to see if the ElemType is "a." If the element is an anchor, the code extracts the URL and directs it into a container for later retrieval.

```
Private Sub HTML1_DoNewElement(ByVal ElemType As String, ByVal EndTag As Boolean, ByVal Attrs As HTMLAttrs, ByVal Text As String, EnableDefault As Boolean)
```

```
    ' gUrls is a global string variable.
    If ElemType = "a" Then
        ' Extract the URLs. (code not shown)
        ' and put them into the variable gUrls.
    End If
```

```
End Sub
```

Extract Attributes Using the For Each Statement with Each Member of the Collection

Note If you do not know what an HTML attribute is, see "About HTML Elements and Attributes," found at the end of this topic.

Attributes can be divided into two simple parts: the name, and the value. The **HTML** control stores the two parts together as properties of the **HTMLAttr** object: the **Name** property and the **Value** property. Thus there is a one-to-one correlation between a single attribute and a single **HTMLAttr** object. To handle more than one attribute, the **HTMLAttr** object is part of a collection of objects—the **HTMLAttrs** collection.

You may now notice that the DoNewElement event contains a reference (called "Attrs") to the **HTMLAttrs** collection. With this reference you can further test for the existence of attributes by using the **Count** property of the collection. And if any attributes are present in the element, the code can insert the attribute into the variable gUrls. This is shown in the code below:

```
If Attrs.Count > 0 Then
    For Each HTMLAttr In Attrs
        gUrls = gUrls & HTMLAttr.Name & "=" & _
            HTMLAttr.Value & Chr(10) & Chr(13)
    Next
End If
```

Combined with the previous code, we have a handy way of first testing if an element is of element type "a," then testing for URL strings and placing them into a text box. The whole code is shown below:

```
Private Sub HTML1_DoNewElement(ByVal ElemType As String, ByVal EndTag As Boolean, ByVal Attrs As HTMLAttrs, ByVal Text As String, EnableDefault As Boolean)
```

```
    ' gUrls is a global string variable.
    If ElemType = "a" Then ' it's an anchor.
        If Attrs.Count > 0 Then ' get attributes (URLs).
            For i = 1 To Attrs.Count
                gUrls = gUrls & Attrs(i).Name & _
                    "=" & Attrs(i).Value & vbCrLf
            Next i
        End If
    End If
End Sub
```

Write the Data to a Text File Using Open and Print Statements in the EndRetrieval Event

After the DoNewElement has parsed every element in the document, the EndRetrieval event occurs. Use this event to write the data into a file, and to reset the variable, as shown below:

```
Private Sub HTML1_EndRetrieval()  
    ' After the retrieval ends, write the data to a  
    ' text file.  
    Open "UrlText.txt" For Output As #1 ' Open file.  
    Print #1, x ' Write to the file.  
    Close #1 ' Close the file.  
    gUrls = "" ' Reset gUrls to null string.  
End Sub
```

Alternatively, you can use the BeginRetrieval event, which occurs before the first DoNewElement event occurs, to open a file for input. In the DoNewElement event, use the **Print** statement to write each URL to the file. Finally, in the EndRetrieval event, close the file.

End Result: a List of URLs

The final result of parsing the document, might look something like this:

```
HREF=/samples/tour/todo.htm  
HREF=/samples/IMAGES/h_samp.map  
HREF=http://www.microsoft.com
```

About HTML Elements and Attributes

To fully program the **HTML** control, it's helpful to understand a little about how the HyperText Markup Language is structured. An HTML document is composed of *elements*. Two few examples are shown below:

```
<IMG SRC="EXPLOSN.GIF">  
<a HREF="/support/search/name.htm">
```

Elements are categorized by *element types* which inform the browser of the kind of information that is contained in the element. The examples above include an image type ("IMG"), and an *anchor* ("a"). An anchor denotes a URL that links to another HTML document.

Furthermore, the element can be broken down into *attributes*. In the examples above, the attributes are:

```
SRC="Explsn.GIF"  
HREF="/support/search/name.htm"
```

While the element type informs the browser of the kind of information of contained in the element, the attribute contains the actual, critical information. The two attributes can finally be broken down into two parts, a *name* and a *value*. The name of the first attribute is "SRC," and it's value is "Explsn.GIF." The second example, a hyperlink reference, has the name "HREF," and the value "support/search/name.htm."

In the two elements shown above, there is only one attribute per element. However, it is possible for a detail to have several attributes, as seen below:

```
<TABLE BORDER=0 CELLPADDING=4 CELLSPACING=0 WIDTH=100%>
```

The table has four attributes: border, cellpadding, cellspacing, and width.

HTML Control

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjHTMLControlC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbobjHTMLControlX":1} {ewc
HLP95EN.DLL,DYNALINK,"Properties":"vbobjHTMLControlP;vbproDragIcon;vbproDragMode;vbproFont;vbproHeight;vbproHel
pContextID;vbproIndex;vbproLeft;vbproName;vbproObjectExt;vbproParent;vbproTabIndex;vbproTabStop;vbproTag;vbproVisibl
e"} {ewc
HLP95EN.DLL,DYNALINK,"Methods":"vbmthDrag;vbmthMove;vbmthSetFocus;vbmthZOrder;vbobjHTMLControlM"}
{ewc
HLP95EN.DLL,DYNALINK,"Events":"vbevtClick;vbevtDbtClick;vbevtDragDrop;vbevtDragOver;vbevtGotFocus;vbevtKeyDown;v
bevtKeyPress;vbevtLostFocus;vbevtMouseDown;vbevtMouseMove;vbobjHTMLControlE"} {ewc
HLP95EN.DLL,DYNALINK,"Specifics":"vbobjHTMLControlS"}
```

The **HTML** Control provides parsing and layout of HTML data, as well as a scrollable view of the selected HTML page. The control supports HTML version 2.0.

Remarks

The **HTML** control lets you implement an HTML viewer, with or without automatic network retrieval of HTML documents, into any application.

Properties such as **Heading1Font** and **VisitedColor** allow the developer to customize the interface to accommodate special needs of end users. Events such as the BeginRetrieval, EndRetrieval, and LayoutComplete further allow you to program features such as progress bars into the application.

The **HTML** control can also be used as a non-visual HTML parser to analyze or process HTML documents. Using the DoNewElement event and the DoRequestSubmit, attributes such as URLs can be easily extracted from HTML documents.

Timeout Property (HTML Control)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproHTMLTimeoutC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbproHTMLTimeoutX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbproHTMLTimeoutA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHTMLTimeoutS"}

Returns or sets the time-out interval (in seconds) for initiating the request for documents. The Timeout event occurs if no data is received within timeout.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object.Timeout</i> [= <i>long</i>]
Visual FoxPro	<i>Object.Timeout</i> [= <i>nTimeout</i>]
Microsoft Visual C++	void <i>dialogclass::OnTimeoutControl</i> ();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	30 seconds	Long
Visual FoxPro	30 seconds	<i>nTimeout</i>

Remarks

Although the Timeout value applies to all document retrieval, the Timeout event is activated only for the main document, not for embedded documents.

Cancel Method (HTML Control)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthCancelHTMLC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbmthCancelHTMLX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbmthCancelHTMLA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthCancelHTMLS"}

Used to terminate document retrieval (including embedded documents), and optionally output a message at the end of the partially retrieved HTML page.

Return Value

Void

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object.Cancel</i> [<i>Message</i>]
Visual FoxPro	<i>Object.Cancel</i> ([<i>cMessage</i>])
Microsoft Visual C++	void Cancel (const VARIANT& <i>Message</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

Development Tool	Argument	Data Type	Description
Microsoft Access and Visual Basic	<i>Message</i>	String	Optional. Message to be appended to the HTML page.
Visual FoxPro	<i>cMessage</i>	Character expression	Optional. Message to be appended to the HTML page.
Visual C++	<i>Message</i>	VARIANT	Optional. Message to be appended to the HTML page.

Remarks

If a message is specified, it will be enclosed in HTML tags, as shown here, and appended to the bottom of the page:

```
<HR><H2>Message</H2>
```

HTML tags are also allowed in the body of the Message.

BeginRetrieval Event

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtBeginRetrievalEventC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"vbevtBeginRetrievalEventX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"vbevtBeginRetrievalEventA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtBeginRetrievalEventS"}
```

Occurs when document retrieval begins.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> _BeginRetrieval
Visual FoxPro	PROCEDURE <i>Object</i> .BeginRetrieval
Microsoft Visual C++	void <i>dialogclass</i> ::OnBeginRetrievalControl();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

None.

Remarks

If the application uses a progress bar, it can be initialized at this time. The **URL** property is copied from the **RequestURL** property immediately before the event occurs.

DoNewElement Event

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtdoNewElementEventC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbevtdoNewElementEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbevtdoNewElementEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtdoNewElementEvents"}
```

Occurs during HTML parsing when a new element is added.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> DoNewElement (<i>ElemType</i> As String , <i>EndTag</i> As Boolean , <i>Attrs</i> As HTMLAttrs , <i>Text</i> as String , <i>EnableDefault</i> As Boolean)
Visual FoxPro	PROCEDURE <i>Object</i> .DoNewElement LPARAMETERS <i>cElemType</i> , <i>lEndTag</i> , <i>Attrs</i> , <i>cText</i> , <i>lEnableDefault</i>
Visual C++	void <i>dialogclass</i> ::OnDoNewElementControl(LPCTSTR <i>ElemType</i> , BOOL <i>EndTag</i> , LPDISPATCH <i>Attrs</i> , LPCTSTR <i>Text</i> , BOOL FAR* <i>EnableDefault</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

Development Tool	Argument	Data Type	Description
Microsoft Access and Visual Basic	<i>Elemtype</i>	String	Element type for tags; empty string for character data.
	<i>EndTag</i>	Boolean	True if an end tag; False otherwise.
	<i>Attrs</i>	HTMLAttrs	Collection of tag attributes. For more information, see HTMLAttrs Collection topic.
	<i>Text</i>	String	Character data; empty for tags.
	<i>EnableDefault</i>	Boolean	Overrides default processing. True indicates default processing, False means override defaults. If <i>EnableDefault</i> is set to false, the HTML control does not store data for this element, but continues parsing.
Visual FoxPro	<i>cElemType</i>	Character	Element type for tags; empty string for character data.
	<i>lEndTag</i>	Logical	True (.T.) if an end tag; otherwise False (.F.).

	<i>Attrs</i>	HTMLAttrs	Collection of tag attributes. For more information, see HTMLAttrs Collection topic.
	<i>cText</i>	Character	Character data; empty for tags.
	<i>IEnableDefault</i>	Logical	Overrides default processing. True indicates default processing, False means override defaults. If <i>IEnableDefault</i> is set to false, the HTML control does not store data for this element, but continues parsing.
Visual C++	<i>Elemtype</i>	LPCTSTR	Element type for tags; pointer to an empty string for character data.
	<i>EndTag</i>	BOOL	True if an end tag; False otherwise.
	<i>Attrs</i>	LPDISPATCH	A pointer to an OLE IDispatch object representing a collection of tag attributes.
	<i>Text</i>	LPCTSTR	A pointer to character data; empty for tags.
	<i>EnableDefault</i>	BOOL	Overrides default processing. True indicates default processing, False means override defaults. If <i>EnableDefault</i> is set to false, the HTML control does not store data for this element, but continues parsing.

Remarks

For character data, ElemType will be an empty string, and Text will contain the character data. For tags, the ElemType will contain the tag type, and the new element's attribute information can be retrieved using the **HTMLAttrs** collection argument.

Some common ElemType values are shown below:

ElemType	Description
a	Anchor. A URL.
font	Font attributes.
img	Image URL and attributes.

DoNewElement Event, HTMLAttrs Collection Example

The example first determines if the **Count** property of the **HTMLAttrs** collection is greater than 0. If so, the **ElemType** argument is placed in a variable along with all attributes of the element. To run the example, place an **HTML** control and a **TextBox** control on a form. Set the **MultiLine** Property of the **TextBox** control to **True**, and set its **ScrollBars** property to Vertical (2). Paste the code into the Declarations section and press F5. Click on the form, and type in a valid URL.

```
Option Explicit
```

```
Private strX As String ' Global variable.
```

```
Private Sub HTML1_DoNewElement(ByVal ElemType As _  
String, ByVal EndTag As Boolean, ByVal Attrs As _  
HTMLAttrs, ByVal Text As String, EnableDefault As Boolean)
```

```
    If Attrs.Count > 0 Then
```

```
        strX = strX & "ElemType: " & ElemType & vbCrLf
```

```
        For Each HTMLAttr in Attrs
```

```
            strX = strX & HTMLAttr.Name & "=" & _
```

```
            HTMLAttr.Value & vbCrLf
```

```
        Next
```

```
        strX = strX & vbCrLf ' add a line at the end
```

```
    End If
```

```
End Sub
```

```
Private Sub Form_Click()
```

```
    Dim strURL as string
```

```
    HTML1.ElemNotification = True ' This must be true.
```

```
    strURL = InputBox("Please type in a valid URL")
```

```
    HTML1.RequestDoc strURL ' Get the document.
```

```
End Sub
```

```
Private Sub HTML1_EndRetrieval()
```

```
    ' Show the text.
```

```
    Text1.Text = strX
```

```
End Sub
```

DoRequestDoc Event

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtdoRequestDocEventC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbevtdoRequestDocEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbevtdoRequestDocEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtdoRequestDocEventS"}
```

Occurs when the user chooses a link to a different URL or when the **RequestDoc** method is called.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> DoRequestDoc (<i>URL As String, Element As HTMLElement, DocInput As DocInput, EnableDefault As Boolean</i>)
Visual FoxPro	PROCEDURE <i>Object</i> .DoRequestDoc LPARAMETERS <i>cURL, Element, DocInput, IEnableDefault</i>
Visual C++	void <i>dialogclass::</i> OnDoRequestDocControl(LPCTSTR <i>URL, LPDISPATCH Element,</i> LPDISPATCH DocInput, BOOL FAR* <i>EnableDefault</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

Development Tool	Argument	Data Type	Description
Microsoft Access and Visual Basic	<i>URL</i>	String	Identifies the requested document
	<i>Element</i>	HTMLElement	Currently unused, but in the future it will identify the anchor element of the link selected by the user.
	<i>DocInput</i>	DocInput	Causes the control to accept input from another source.
	<i>EnableDefault</i>	Boolean	Overrides default processing. True indicates default processing, False means cancel default processing.
Visual FoxPro	<i>cURL</i>	Character	Identifies the requested document
	<i>Element</i>		Currently unused, but in the future it will identify the anchor element of the link selected by the user.
	<i>DocInput</i>		Causes the control to accept input from

	<i>IEnableDefault</i>	Logical	another source. Overrides default processing. True (.T.) indicates default processing, False (.F.) means cancel default processing.
Visual C++	<i>URL</i>	LPCTSTR	Pointer to a string that identifies the requested document.
	<i>Element</i>	LPDISPATCH	Currently unused, but in the future it will be a pointer to an OLE IDispatch object representing the anchor element of the link selected by the user.
	<i>DocInput</i>	LPDISPATCH	A pointer to an OLE IDispatch object that causes the control to accept input from another source.
	<i>EnableDefault</i>	BOOL	Overrides default processing. True indicates default processing, False means cancel default processing.

Default Action

The default action of DoRequestDoc depends on the URL type.

URL Type	Default Action
HTTP and File URL	DoRequestDoc creates a default DocInput object for retrieving the document
Other URL types or from a different source for any URL type	The DocInput property is set during event handling

Remarks

When **RequestDoc** is invoked, the DoRequestDoc event occurs to determine if the **DocInput** object can be used for retrieval. The **RequestURL** property is then set to the URL parameter specified. The **URL** property is not updated until retrieval is successfully underway and the BeginRetrieval event is activated.

If the event is not canceled, the **RequestURL** property is set by the control. The **URL** property is not updated until retrieval is successfully underway and the BeginRetrieval event is activated.

DoRequestEmbedded Event

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":vbEvtDoRequestEmbeddedEventC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":vbEvtDoRequestEmbeddedEventX:1} {ewc
HLP95EN.DLL,DYNALINK,"Applies To":vbEvtDoRequestEmbeddedEventA"} {ewc
HLP95EN.DLL,DYNALINK,"Specifics":vbEvtDoRequestEmbeddedEventS"}
```

Occurs when an embedded document, such as an image, is to be retrieved for inline display.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> DoRequestEmbedded (ByVal URL As String, ByVal Element As HTMLElement, ByVal DocInput As DocInput, EnableDefault As Boolean)
Visual FoxPro	PROCEDURE <i>Object</i> .DoRequestEmbedded LPARAMETERS <i>cURL</i> , <i>Element</i> , <i>DocInput</i> , <i>EnableDefault</i>
Visual C++	void dialogclass::OnDoRequestEmbeddedControl(LPCTSTR URL, LPDISPATCH Element, LPDISPATCH DocInput, BOOL FAR* EnableDefault);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

Development Tool	Argument	Data Type	Description
Microsoft Access and Visual Basic	<i>URL</i>	String	Identifies the requested document.
	<i>Element</i>	HTMLElement	Currently unused, but in the future it will identify the HTML element of the embedded document.
	<i>DocInput</i>	DocInput	Causes the control to accept input from another source.
	<i>EnableDefault</i>	Boolean	Overrides default processing. True indicates default processing, False means cancel the request.
Visual FoxPro	<i>cURL</i>	Character	Identifies the requested document.
	<i>Element</i>		Currently unused, but in the future it will identify the HTML element of the embedded

	<i>DocInput</i>		document. Causes the control to accept input from another source.
	<i>IEnableDefault</i>	Logical	Overrides default processing. True (.T.) indicates default processing, False (.F.) means cancel the request.
Visual C++	<i>URL</i>	LPCTSTR	Pointer to a string that identifies the requested document.
	<i>Element</i>	LPDISPATCH	Currently unused, but in the future it will be a pointer to an OLE IDispatch object representing the HTML element of the embedded document.
	<i>DocInput</i>	LPDISPATCH	A pointer to an OLE IDispatch object that causes the control to accept input from another source.
	<i>EnableDefault</i>	BOOL	Overrides default processing. True indicates default processing, False means cancel the request.

Default Action

The default action of DoRequestEmbedded depends on the URL type.

URL Type	Default Action
HTTP and File URL	DoRequestEmbedded creates a default DocInput object for retrieving the document.
Other URL types or from a different source for any URL type	The DocInput property is set during event handling.

DoRequestSubmit Event

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtDoRequestSubmitEventC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbevtDoRequestSubmitEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbevtDoRequestSubmitEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtDoRequestSubmitEventS"}
```

Occurs when the user selects form submission, or when the **RequestSubmit** method of the Form is called.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> DoRequestSubmit (ByVal <i>URL</i> As String , ByVal <i>Form</i> As HTMLForm , ByVal <i>DocOutput</i> As DocOutput , <i>EnableDefault</i> As Boolean)
Visual FoxPro	PROCEDURE <i>Object</i> .DoRequestSubmit LPARAMETERS <i>cURL</i> , <i>Form</i> , <i>DocOutput</i> , <i>IEnableDefault</i>
Visual C++	void <i>dialogclass</i> ::OnDoRequestSubmitControl(LPCTSTR <i>URL</i> , LPDISPATCH <i>Form</i> , LPDISPATCH <i>DocOutput</i> , BOOL FAR* <i>EnableDefault</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

Microsoft Access and Visual Basic

Argument	Data Type	Description
<i>URL</i>	String	Identifies the action URL for that form, and includes the search string for GET form methods as described in Remarks.
<i>Form</i>	HTMLForm	Identifies the form being submitted, and is an item in the Forms collection
<i>DocOutput</i>	DocOutput	Causes output to another target.
<i>EnableDefault</i>	Boolean	Overrides default processing. True indicates default processing, False means override defaults. To cancel the submission request, set the EnableDefault argument to False. If the event is not canceled, the RequestURL property is set by the control. The URL property is not updated until retrieval is successfully underway and the BeginRetrieval event occurs.

Visual FoxPro

<i>cURL</i>	Character	Identifies the action URL for that form, and includes the search string for GET form methods as described in Remarks.
<i>Form</i>		Identifies the form being submitted, and is an item in the Forms collection.

<i>DocOutput</i>		Causes output to another target.
<i>!EnableDefault</i>	Logical	Overrides default processing. True (.T.) indicates default processing, False (.F.) means override defaults. To cancel the submission request, set the <i>EnableDefault</i> argument to False . If the event is not canceled, the RequestURL property is set by the control. The URL property is not updated until retrieval is successfully underway and the <i>BeginRetrieval</i> event occurs.

Visual C++

<i>URL</i>	LPCTSTR	Pointer to a string that identifies the action URL for that form, and includes the search string for GET form methods as described in Remarks.
<i>Form</i>	LPDISPATCH	Pointer to an OLE IDispatch object representing the form being submitted.
<i>DocOutput</i>	LPDISPATCH	Pointer to an OLE IDispatch object that causes output to another target.
<i>EnableDefault</i>	BOOL	Overrides default processing. True indicates default processing, False means override defaults. To cancel the submission request, set the <i>EnableDefault</i> argument to False . If the event is not canceled, the RequestURL property is set by the control. The URL property is not updated until retrieval is successfully underway and the <i>BeginRetrieval</i> event occurs.

Default Action

The default action of *DoRequestSubmit* is to output the form's contents using HTTP, and input the reply as the next main document. To submit form data using a different source and/or target during event handling, you can modify the **DocOutput** property to specify some other target and link the **DocInput** property to receive the reply. To submit form data to another target without receiving the reply in the HTML control, modify the **DocOutput** property to some other target and unlink the **DocInput** property so that the reply document is discarded.

Remarks

Currently, the form contents for submission always consist of URL-encoded field values contained in the **HTMLForm** object's **URLEncodedBody** property. In the future, multipart content data will also be submitted for file uploading. If the form's submission method is GET (rather than POST), the string passed in the URL parameter of this event will have the URL-encoded body appended after the search character (question mark).

EndRetrieval Event

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevEndRetrievalEventC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"vbevEndRetrievalEventX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"vbevEndRetrievalEventA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevEndRetrievalEventS"}
```

Occurs when document retrieval, including embedded documents to be displayed inline, is complete.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> _EndRetrieval
Visual FoxPro	PROCEDURE <i>Object</i> .EndRetrieval
Microsoft Visual C++	void <i>dialogclass</i> ::OnEndRetrievalControl();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

None.

Remarks

Use this event to terminate a progress bar update.

LayoutComplete Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevLayoutCompleteEventC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbevLayoutCompleteEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbevLayoutCompleteEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevLayoutCompleteEventS"}

Occurs when layout of the HTML document is complete.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> _LayoutComplete
Visual FoxPro	PROCEDURE <i>Object</i> .LayoutComplete
Microsoft Visual C++	void <i>dialogclass</i> ::OnLayoutCompleteControl();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

None.

Remarks

Embedded document retrieval may not be complete, however, the size of each embedded document and the position of all elements is determined.

ParseComplete Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtParseCompleteEventC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbevtParseCompleteEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbevtParseCompleteEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtParseCompleteEvents"}

Occurs when parsing of the HTML document is complete.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> _ParseComplete()
Visual FoxPro	PROCEDURE <i>Object</i> .ParseComplete
Microsoft Visual C++	void <i>dialogclass</i> ::OnParseCompleteControl();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

None.

Remarks

Layout and embedded document retrieval may not be complete.

UpdateRetrieval Event

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevUpdateRetrievalEventC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"vbevUpdateRetrievalEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"vbevUpdateRetrievalEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevUpdateRetrievalEventS"}
```

Occurs periodically as the document and embedded objects are retrieved.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> _UpdateRetrieval()
Visual FoxPro	PROCEDURE <i>Object</i> .UpdateRetrieval
Microsoft Visual C++	void <i>dialogclass</i> ::OnUpdateRetrievalControl();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

None.

Remarks

The **RetrieveBytesTotal** and **RetrieveBytesDone** properties can be queried at the time this event is activated to update a progress bar.

RequestAllEmbedded Method

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthRequestAllEmbeddedMethodC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"vbmthRequestAllEmbeddedMethodX":1} {ewc  
HLP95EN.DLL,DYNALINK,"Applies To":"vbmthRequestAllEmbeddedMethodA"} {ewc  
HLP95EN.DLL,DYNALINK,"Specifics":"vbmthRequestAllEmbeddedMethodS"}
```

Requests retrieval of all embedded documents via the DoRequestEmbedded event.

Return Value

Void

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .RequestAllEmbedded
Visual FoxPro	<i>Object</i> .RequestAllEmbedded()
Microsoft Visual C++	void RequestAllEmbedded ();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

None.

Remarks

This method is used in conjunction with the **DeferRetrieval** property to control inline display of embedded documents.

RequestDoc Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthRequestDocMethodC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthRequestDocMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthRequestDocMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthRequestDocMethodS"}

Requests retrieval of a new main document identified by the URL.

Return Value

Void

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .RequestDoc <i>URL</i>
Visual FoxPro	<i>Object</i> .RequestDoc(<i>cURL</i>)
Microsoft Visual C++	void RequestDoc(LPCTSTR <i>URL</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

Development Tool	Argument	Data Type	Description
Microsoft Access and Visual Basic	<i>URL</i>	String	Identifies the new main document to be retrieved.
Visual FoxPro	<i>cURL</i>	Character	Identifies the new main document to be retrieved.
Visual C++	<i>URL</i>	Pointer to a null-terminated character string	Identifies the new main document to be retrieved.

Remarks

When **RequestDoc** is invoked, the DoRequestDoc event occurs to determine the **DocInput** object to be used for retrieval. The **RequestURL** property is then set to the specified URL argument. The **URL** property is not updated until retrieval is successfully underway and the BeginRetrieval event occurs.

RequestDoc Method Example

The example requests an HTML document using the URL of the requested document. To try this example, place an **HTML** control on a form and paste the code into the Load event of the form.

```
Private Sub Form_Load()  
    HTML1.RequestDoc "HTTP://www.microsoft.com"  
End Sub
```

RequestSubmit Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthRequestSubmitMethodC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthRequestSubmitMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthRequestSubmitMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthRequestSubmitMethodS"}

Requests submission of a form.

Return Value

Void

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> . RequestSubmit
Visual FoxPro	<i>Object</i> .RequestSubmit()

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Arguments

None.

Remarks

When **RequestSubmit** is invoked, the DoRequestSubmit event occurs to determine the target document to be used for submission. The **RequestURL** property is then set to the action URL of the form. The **URL** property is not updated until retrieval is successfully underway and the BeginRetrieval event occurs.

BackColor Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBackColorPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproBackColorPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproBackColorPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproBackColorPropertyS"}

Returns or sets the URL of an image to be used as the background image of the document.
Read/write and available and design time and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .BackColor [= <i>string</i>]
Visual FoxPro	<i>Object</i> .BackColor[= <i>cExpression</i>]
Microsoft Visual C++	CString GetBackColor(); void SetBackColor(LPCTSTR <i>lpszNewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Empty string	String
Visual FoxPro	Empty string	Character
Visual C++	Empty object	CString object

Remarks

Can be overridden by the background image of the document (<BODY BACKGROUND=...>) if this attribute is present and the **UseDocColors** property is **True**. The background image is tiled to fill the view area of the control window.

BaseURL Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBaseURLPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproBaseURLPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproBaseURLPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproBaseURLPropertyS"}

Returns the URL of the <BASE> element of the current document, used for relative URL resolution. If no <BASE> element exists in the document, this property is the same as the **URL** property. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> . BaseURL
Visual FoxPro	<i>Object</i> .BaseURL
Microsoft Visual C++	CString GetBaseURL ();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Empty string	String
Visual FoxPro	Empty string	Character
Visual C++	Empty object	CString object

Remarks

If no <BASE> element exists in the document, this property is the same as the **URL** property

DeferRetrieval Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproDeferRetrievalPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproDeferRetrievalPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproDeferRetrievalPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproDeferRetrievalPropertyS"}

Returns or sets a value to turn inline retrieval of embedded documents on or off. Read/write and available at design time and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .DeferRetrieval [= <i>boolean</i>]
Visual FoxPro	<i>Object</i> .DeferRetrieval[= <i>IExpression</i>]
Visual C++	BOOL GetDeferRetrieval(); void SetDeferRetrieval(BOOL <i>bNewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	False	Boolean
Visual FoxPro	False (.F.)	Logical

Remarks

You can set this property to turn inline retrieval of embedded documents off or on. If you implement caching, set **DeferRetrieval** to **False** to always display cached documents inline.

DocBackColor Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproDocBackColorPropertyC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbproDocBackColorPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbproDocBackColorPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproDocBackColorPropertyS"}

Returns the document background color. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .DocBackColor
Visual FoxPro	<i>Object</i> .DocBackColor
Visual C++	unsigned long GetDocBackColor();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	BackColor	Long
Visual FoxPro	BackColor value	Numeric
Visual C++	BackColor value	unsigned long

Remarks

This property corresponds to the BGCOLOR attribute of the BODY tag. If this attribute is not present, HTML defaults to the value of the **BackColor** property.

DocForeColor Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproDocForeColorPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproDocForeColorPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproDocForeColorPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproDocForeColorPropertyS"}

Returns the document foreground (text) color. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .DocForeColor
Visual FoxPro	<i>Object</i> .DocForeColor
Visual C++	unsigned long GetDocForeColor();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	ForeColor	Long
Visual FoxPro	ForeColor value	Numeric
Visual C++	ForeColor value	unsigned long

Remarks

This property corresponds to the TEXT attribute of the BODY tag. If this attribute is not present, HTML defaults to the value of the **ForeColor** property.

DocLinkColor Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproDocLinkColorPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproDocLinkColorPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproDocLinkColorPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproDocLinkColorPropertyS"}

Returns a document link color. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .DocLinkColor
Visual FoxPro	<i>Object</i> .DocLinkColor
Visual C++	unsigned long GetDocLinkColor();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	LinkColor	Long
Visual FoxPro	LinkColor value	Numeric
Visual C++	LinkColor value	unsigned long

Remarks

This property corresponds to the LINK attribute of the BODY tag. If this attribute is not present, HTML defaults to the value of the **LinkColor** property.

DocVisitedColor Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproDocVisitedColorPropertyC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbproDocVisitedColorPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbproDocVisitedColorPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproDocVisitedColorPropertyS"}

Returns a document visited link color. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .DocVisitedColor
Visual FoxPro	<i>Object</i> .DocVisitedColor
Visual C++	unsigned long GetDocVisitedColor();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	VisitedColor	Long
Visual FoxPro	VisitedColor value	Numeric
Visual C++	VisitedColor value	unsigned long

Remarks

This property corresponds to the VLINK attribute of the BODY tag. If this attribute is not present, HTML defaults to the value of the **VisitedColor** property.

ElemNotification Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproElemNotificationPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproElemNotificationPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproElemNotificationPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproElemNotificationPropertyS"}

Returns or sets a value to trigger the DoNewElement event during HTML parsing. Read/write and available at design time and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .ElemNotification
Visual FoxPro	<i>Object</i> .ElemNotification[= <i>IExpression</i>]
Visual C++	BOOL GetElemNotification(); void SetElemNotification(BOOL <i>bNewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	False	Boolean
Visual FoxPro	False (.F.)	Logical

Remarks

You can set this property to **True** when using the HTML control as a (visual or nonvisual) parser.

See Also

DoNewElement event

FixedFont Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproFixedFontPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFixedFontPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproFixedFontPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFixedFontPropertyS"}

Returns or sets a value that indicates the font for fixed-width text. Read/write and available at design and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object.FixedFont</i> [=string]
Visual FoxPro	<i>Object.FixedFont</i> [= oExpression]
Visual C++	COleFont GetFixedFont (); void SetFixedFont (LPDISPATCH <i>NewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Courier New, size 10.	String
Visual FoxPro	Courier New, size 10	Object
Visual C++	Courier New, size 10	COleFont object

Forms Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproFormsPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFormsPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproFormsPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFormsPropertyS"}

Returns a reference to the **HTMLForms** collection. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object.Forms</i>
Visual FoxPro	<i>Object.Forms</i>
Visual C++	CHTMLForms GetForms ();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	None	HTMLForms

Visual FoxPro	None	Object
Visual C++	None	CHTMLForms object

Remarks

This property can be indexed directly to call the default **Item** method.

Heading1Font Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproHeading1FontPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproHeading1FontPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproHeading1FontPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHeading1FontPropertyS"}

Returns or sets the font for heading level 1 text (<H1> elements). Read/write and available at design time and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .Heading1Font [=string]
Visual FoxPro	<i>Object</i> .Heading1Font[= oExpression]
Visual C++	COleFont GetHeading1Font(); void SetHeading1Font(LPDISPATCH <i>NewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Times New Roman, size 24, Bold	String
Visual FoxPro	Times New Roman, size 24, Bold	Object
Visual C++	Times New Roman, size 24, Bold	COleFont object

Heading2Font Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproHeading2FontPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproHeading2FontPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproHeading2FontPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHeading2FontPropertyS"}

Returns or sets the font for heading level 2 text (<H2> elements). Read/write and available at design time and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .Heading2Font [=string]
Visual FoxPro	<i>Object</i> .Heading2Font[= oExpression]
Visual C++	COleFont GetHeading2Font(); void SetHeading2Font(LPDISPATCH <i>NewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Times New Roman, size 18, Bold	String
Visual FoxPro	Times New Roman, size 18, Bold	Object
Visual C++	Times New Roman, size 18, Bold	COleFont object

Heading3Font Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproHeading3FontPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproHeading3FontPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproHeading3FontPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHeading3FontPropertyS"}

Returns or sets the font for heading level 3 text (<H3> elements). Read/write and available at design and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .Heading3Font [=string]
Visual FoxPro	<i>Object</i> .Heading3Font[= oExpression]
Visual C++	COleFont GetHeading3Font(); void SetHeading3Font(LPDISPATCH <i>NewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Times New Roman, size 14, Bold	String
Visual FoxPro	Times New Roman, size 14, Bold	Object
Visual C++	Times New Roman, size 14, Bold	COleFont object

Heading4Font Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproHeading4FontPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproHeading4FontPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproHeading4FontPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHeading4FontPropertyS"}

Returns or sets the font for heading level 4 text (<H4> elements). Read/write and available at design and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .Heading4Font [=string]
Visual FoxPro	<i>Object</i> .Heading4Font[= <i>oExpression</i>]
Visual C++	COleFont GetHeading4Font(); void SetHeading4Font(LPDISPATCH <i>NewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Times New Roman, size 12, Bold	String
Visual FoxPro	Times New Roman, size 12, Bold	Object
Visual C++	Times New Roman, size 12, Bold	COleFont object

Heading5Font Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":":vbproHeading5FontPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":":vbproHeading5FontPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":":vbproHeading5FontPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":":vbproHeading5FontPropertyS"}

Returns or sets the font for heading level 5 text (<H5> elements). Read/write and available at design and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .Heading5Font [=string]
Visual FoxPro	<i>Object</i> .Heading5Font[= <i>oExpression</i>]
Visual C++	COleFont GetHeading5Font(); void SetHeading5Font(LPDISPATCH <i>NewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Times New Roman, size	String

Visual FoxPro	10, Bold Times New Roman, size 10, Bold	Object
Visual C++	Times New Roman, size 10, Bold	COleFont object

Heading6Font Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproHeading6FontPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproHeading6FontPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproHeading6FontPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHeading6FontPropertyS"}

Returns or sets the font for heading level 6 text (<H6> elements). Read/write and available at design and run time.

Syntax

Development Tool Syntax

Microsoft Access and Visual Basic	<i>object</i> .Heading6Font [=string]
Visual FoxPro	<i>Object</i> .Heading6Font[= <i>oExpression</i>]
Visual C++	COleFont GetHeading6Font(); void SetHeading6Font(LPDISPATCH <i>NewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Times New Roman, size 8, Bold	String
Visual FoxPro	Times New Roman, size 8, Bold	Object
Visual C++	Times New Roman, size 8, Bold	COleFont object

LayoutDone Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproLayoutDonePropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproLayoutDonePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproLayoutDonePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproLayoutDonePropertyS"}

Returns a value that indicates whether the layout phase is complete. Read/only and unavailable at design time.

Syntax

Development Tool Syntax

Microsoft Access and Visual Basic *object.LayoutDone*
Visual FoxPro *Object.LayoutDone*
Visual C++ **BOOL GetLayoutDone();**

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	False	Boolean
Visual FoxPro	False (.F.)	Logical

Remarks

This property is set to **False** when document retrieval starts, and set to **True** when layout (placement of items on the page) of the main document is complete.

LinkColor Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproLinkColorPropertyC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbproLinkColorPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbproLinkColorPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproLinkColorPropertyS"}

Returns or sets the default link color. Read/write and available at design time and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .LinkColor [= <i>color</i>]
Visual FoxPro	<i>Object</i> .LinkColor[= <i>nColor</i>]
Visual C++	unsigned long GetLinkColor(); void SetLinkColor(unsigned long <i>newValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Blue (0,0,255)	RGB
Visual FoxPro	Blue (16711680)	Numeric
Visual C++	Blue	unsigned long

Remarks

This property can be overridden by the **DocLinkColor** property if such a document color is present and the **UseDocColors** property is **True**.

ParseDone Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproParseDonePropertyC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"vbproParseDonePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"vbproParseDonePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproParseDonePropertyS"}
```

Returns a value that indicates whether the parsing phase is complete. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .ParseDone
Visual FoxPro	<i>Object</i> .ParseDone
Visual C++	BOOL GetParseDone();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	False	Boolean
Visual FoxPro	False (.F.)	Logical

Settings

The **ParseDone** property settings are:

Setting	Description
True	Parsing is complete.
False	Parsing is incomplete.

Remarks

This property is set to **False** when document retrieval starts, and set to **True** when parsing of the main document is complete.

Redraw Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproRedrawPropertyC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbproRedrawPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbproRedrawPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproRedrawPropertyS"}

Returns or sets a value that indicates whether drawing should occur as data changes or the window is scrolled. Read/write and available at design and run time.

Syntax

Development Tool	Syntax
------------------	--------

Microsoft Access and Visual Basic	<i>object.Redraw</i> [= <i>boolean</i>]
Visual FoxPro	<i>Object.Redraw</i> [= <i>IExpression</i>]
Visual C++	BOOL GetRedraw (); void SetRedraw BOOL <i>bNewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	True	Boolean
Visual FoxPro	True (.T.)	Logical

Remarks

To make changes and avoid flickering (redrawing when each change is made), set the **Redraw** property to **False**, make the changes, and then set it back to **True**. When **Redraw** is set to **True**, the window will be redrawn.

RequestURL Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproRequestURLPropertyC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbproRequestURLPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbproRequestURLPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproRequestURLPropertyS"}

Returns the URL string identifying the new document requested. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .RequestURL
Visual FoxPro	<i>Object</i> .RequestURL
Visual C++	CString GetRequestURL();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Empty string	String
Visual FoxPro	Empty string	Character
Visual C++	Empty object	CString object

Remarks

You can specify this property by calling the RequestDoc method. The property is set by the control during default processing for the DoRequestDoc event.

RetainSource Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproRetainSourcePropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproRetainSourcePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproRetainSourcePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproRetainSourcePropertyS"}

Returns or sets a value that indicates whether source text should be retained and available via the **SourceText** property. Read/write and available at design time and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .RetainSource [= <i>boolean</i>]
Visual FoxPro	<i>Object</i> .RetainSource[= <i>IExpression</i>]
Visual C++	BOOL GetRetainSource(); void SetRetainSource(BOOL <i>bNewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	True	Boolean
Visual FoxPro	True (.T.)	Logical

Remarks

This property can be set to **False** to save memory when you do not need the source text of the main document.

RetrieveBytesDone Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproRetrieveBytesDonePropertyC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbproRetrieveBytesDonePropertyX":1} {ewc
HLP95EN.DLL,DYNALINK,"Applies To":"vbproRetrieveBytesDonePropertyA"} {ewc
HLP95EN.DLL,DYNALINK,"Specifics":"vbproRetrieveBytesDonePropertyS"}

Returns the completed byte size of the objects being retrieved. This property is zero if no retrieval is in progress. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> . RetrieveBytesDone
Visual FoxPro	<i>Object</i> .RetrieveBytesDone
Visual C++	long GetRetrieveBytesDone() ;

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	0	Long
Visual FoxPro	0	Numeric

RetrieveBytesTotal Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproRetrieveBytesTotalPropertyC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbproRetrieveBytesTotalPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbproRetrieveBytesTotalPropertyA"} {ewc
HLP95EN.DLL,DYNALINK,"Specifics":"vbproRetrieveBytesTotalPropertyS"}

Returns the total byte size of the objects to be retrieved, including embedded objects and the document itself. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> . RetrieveBytesTotal
Visual FoxPro	<i>Object</i> .RetrieveBytesTotal
Visual C++	long GetRetrieveBytesTotal() ;

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
------------------	---------------	-----------

Microsoft Access, Visual Basic, and Visual C++	0	Long
Visual FoxPro	0	Numeric

Remarks

If **DeferRetrieval** is set to **True**, **RetrieveBytesTotal** does not include embedded objects. This value can change during retrieval as object sizes are determined. This property is zero if no retrieval is in progress.

SourceText Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproSourceTextPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproSourceTextPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproSourceTextPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproSourceTextPropertyS"}

Returns the source text of the main document. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .SourceText
Visual FoxPro	<i>Object</i> .SourceText
Visual C++	CString GetSourceText();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	None	String
Visual FoxPro	Empty string	Character
Visual C++	None	CString object

Remarks

This property will be empty if the **RetainSource** property is **False** or if no main document has been retrieved.

TotalHeight Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproTotalHeightPropertyC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"vbproTotalHeightPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"vbproTotalHeightPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproTotalHeightPropertyS"}
```

Returns the total height of the document in pixels. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .TotalHeight
Visual FoxPro	<i>Object</i> .TotalHeight
Visual C++	long GetTotalHeight();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	0	Long
Visual FoxPro	0	Numeric

Remarks

This property reflects the total height of the document, including the area that may not be visible because the view is smaller than the document. This property is updated as parsing and layout of the HTML document occurs. Its value is final when the EndRetrieval event is activated.

TotalWidth Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproTotalWidthPropertyC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbproTotalWidthPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbproTotalWidthPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproTotalWidthPropertyS"}

Returns the total width of the document in pixels. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .TotalWidth
Visual FoxPro	<i>Object</i> .TotalWidth
Visual C++	long GetTotalWidth();

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	0	Long
Visual FoxPro	0	Numeric

Remarks

This property reflects the total width of the document, including the area that may not be visible because the view is smaller than the document. This property is updated as parsing and layout of the HTML document occurs. Its value is final when the EndRetrieval event is activated.

UnderlineLinks Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproUnderlineLinksPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproUnderlineLinksPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproUnderlineLinksPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproUnderlineLinksPropertyS"}

Returns or sets a value that indicates whether links should be underlined. Read/write and available at design and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .UnderlineLinks [= <i>boolean</i>]
Visual FoxPro	<i>Object</i> .UnderlineLinks[= <i>IExpression</i>]
Visual C++	BOOL GetUnderlineLinks(); void SetUnderlineLinks(BOOL <i>bNewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	True	Boolean
Visual FoxPro	True	Logical

URLEncodedBody Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproURLEncodedBodyPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproURLEncodedBodyPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproURLEncodedBodyPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproURLEncodedBodyPropertyS"}

Returns the URL-encoded body text, representing the values of all form fields used for HTTP submission. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .URLEncodedBody
Visual FoxPro	<i>Object</i> .URLEncodedBody

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	None	String
Visual FoxPro	Empty string	Character

UseDocColors Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproUseDocColorsPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproUseDocColorsPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproUseDocColorsPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproUseDocColorsPropertyS"}

Returns or sets a value that indicates whether document colors should be used when present.
Read/write and available at design and run time

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .UseDocColors [= <i>boolean</i>]
Visual FoxPro	<i>Object</i> .UseDocColors[= <i>IExpression</i>]
Visual C++	BOOL GetUseDocColors(); void SetUseDocColors(BOOL <i>bNewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	True	Boolean
Visual FoxPro	True	Logical

Remarks

If this property value is **True**, the document colors (if present) override the default colors. For example, if the <BODY LINK=...> attribute is present and **UseDocColors** is **True**, then the color specified for the LINK attribute is used to display active links; otherwise, the **LinkColor** property value is used.

ViewSource Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":""} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproViewSourcePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproViewSourcePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproViewSourcePropertyS"}
```

Returns or sets a value that indicates whether the control should display HTML source as plain text. Read/write and available at design and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .ViewSource [= <i>boolean</i>]
Visual FoxPro	<i>Object</i> .ViewSource[= <i>IExpression</i>]
Visual C++	BOOL GetViewSource(); void SetViewSource(BOOL <i>bNewValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access, Visual Basic, and Visual C++	True	Boolean
Visual FoxPro	True	Logical

Remarks

This property is set to **True** to view the source text of the main document. If this property is **True** and **RetainSource** is **False**, document retrieval is initiated to obtain the source text for viewing.

VisitedColor Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproVisitedColorPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproVisitedColorPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproVisitedColorPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproVisitedColorPropertyS"}

Returns or sets a value that indicates the default visited link color. Read/write and available at design and run time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .VisitedColor [= <i>string</i>]
Visual FoxPro	<i>Object</i> .VisitedColor[= <i>nColor</i>]
Visual C++	unsigned long GetVisitedColor(); void SetVisitedColor(unsigned long <i>newValue</i>);

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	Purple (255,0,255)	RGB
Visual FoxPro	Purple (16711935)	Numeric
Visual C++	Purple	unsigned long

Remarks

This property can be overridden by the **DocVisitedColor** property if such a document color is present and the **UseDocColors** property is **True**.

HTMLAttr Object

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjHTMLAttrObjectC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"vbobjHTMLAttrObjectX":1} {ewc  
HLP95EN.DLL,DYNALINK,"Properties":"vbobjHTMLAttrObjectP"} {ewc  
HLP95EN.DLL,DYNALINK,"Methods":"vbobjHTMLAttrObjectM"} {ewc  
HLP95EN.DLL,DYNALINK,"Events":"vbobjHTMLAttrObjectE"} {ewc  
HLP95EN.DLL,DYNALINK,"Specifics":"vbobjHTMLAttrObjectS"}
```

An **HTMLAttr** object contains the attribute names and values of an HTML element.

HTMLAttrs

Remarks

The **HTMLAttr** object is a member of the **HTMLAttrs** Collection. This collection is available in the DoNewElement event. The following code prints the element names and values of each **HTMLAttr** object in a collection:

```
Private Sub HTML1_DoNewElement(ElemType As String, _  
EndTag As Boolean, Attrs As HTMLAttrs, Text as String, _  
EnableDefault As Boolean)  
  
Dim i As Integer  
For i = 1 to Attrs.Count  
    txtAttributes = Attrs(i).Name & Attrs.Value & vbCrLf  
Next i  
  
End Sub
```

HTMLAttrs Collection

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcolHTMLAttrsCollectionC"}           {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbcolHTMLAttrsCollectionX":1}             {ewc
HLP95EN.DLL,DYNALINK,"Properties":"vbcolHTMLAttrsCollectionP"}           {ewc
HLP95EN.DLL,DYNALINK,"Methods":"vbcolHTMLAttrsCollectionM"}             {ewc
HLP95EN.DLL,DYNALINK,"Events":"vbcolHTMLAttrsCollectionE"}             {ewc
HLP95EN.DLL,DYNALINK,"Specifics":"vbcolHTMLAttrsCollectionS"}           {ewc
```

An **HTMLAttrs** collection contains a collection of **HTMLAttr** objects. An **HTMLAttrs** collection is passed as an argument in the DoNewElement event.

HTMLAttrs



Remarks

A reference to the **HTMLAttrs** collection is found only in the DoNewElement event.

The **HTMLAttrs** collection is a 1-based collection. To enumerate through the collection, you can use the following code:

```
Sub HTML1_DoNewElement(ByVal ElemType As String, ByVal EndTag As Boolean,
ByVal Attrs As HTMLAttrs, ByVal Text As String, EnableDefault As Boolean)
```

```
    Dim i As Integer ' Counter variable.

    For i = 1 to Attrs.Count
        ' Process each HTMLAttr object here.
    Next i
```

```
End Sub
```

Name Property (HTMLAttr Object)

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproHTMLAttrNameC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"vbproHTMLAttrNameX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"vbproHTMLAttrNameA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHTMLAttrNameS"}
```

Returns the attribute name. This string is never empty, and can be uppercase, lowercase or mixed case. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
------------------	--------

Microsoft Access and Visual Basic	<i>object.Name</i>
--------------------------------------	--------------------

Visual FoxPro	<i>Object.Name</i>
---------------	--------------------

Return Values

Development Tool	Default Value	Data Type
------------------	---------------	-----------

Microsoft Access and Visual Basic	None	String
--------------------------------------	------	--------

Visual FoxPro	None	Character
---------------	------	-----------

Value Property (HTMLAttr Object)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproHTMLAttrValueC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbproHTMLAttrValueX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"vbproHTMLAttrValueA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHTMLAttrValueS"}

Returns the attribute value. This string can be empty. If not empty, the string is unescaped (decoded).
Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
-------------------------	---------------

Microsoft Access and Visual Basic	<i>object.Value</i>
--------------------------------------	---------------------

Visual FoxPro	<i>Object.Value</i>
---------------	---------------------

Return Values

Development Tool	Default Value	Data Type
-------------------------	----------------------	------------------

Microsoft Access and Visual Basic	None	String
--------------------------------------	------	--------

Visual FoxPro	None	Character
---------------	------	-----------

HTMLForms Collection

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcolHTMLFormsCollectionC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"vbcolHTMLFormsCollectionX":1} {ewc  
HLP95EN.DLL,DYNALINK,"Properties":"vbcolHTMLFormsCollectionP"} {ewc  
HLP95EN.DLL,DYNALINK,"Methods":"vbcolHTMLFormsCollectionM"} {ewc  
HLP95EN.DLL,DYNALINK,"Events":"vbcolHTMLFormsCollectionE"} {ewc  
HLP95EN.DLL,DYNALINK,"Specifics":"vbcolHTMLFormsCollectionS"}
```

An **HTMLForms** collection contains a collection of **HTMLForm** objects. The **Forms** property of the **HTML** control returns a reference to the **HTMLForms** collection.

HTMLForms



HTMLForm Object

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjHTMLFormObjectC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"vbobjHTMLFormObjectX":1} {ewc  
HLP95EN.DLL,DYNALINK,"Properties":"vbobjHTMLFormObjectP"} {ewc  
HLP95EN.DLL,DYNALINK,"Methods":"vbobjHTMLFormObjectM"} {ewc  
HLP95EN.DLL,DYNALINK,"Events":"vbobjHTMLFormObjectE"} {ewc  
HLP95EN.DLL,DYNALINK,"Specifics":"vbobjHTMLFormObjectS"}
```

An **HTMLForm** object contains information for submitting documents using the HTTP protocol.



URL Property (HTMLForm Object)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproHTMLFormURLC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproHTMLFormURLX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproHTMLFormURLA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHTMLFormURLS"}

Returns the action URL for the form. Read-only and unavailable at design time.

Syntax

Development Tool	Syntax
Microsoft Access and Visual Basic	<i>object</i> .URL
Visual FoxPro	<i>Object</i> .URL
Visual C++	CString GetURL();

Return Values

Development Tool	Default Value	Data Type
Microsoft Access and Visual Basic	None	String
Visual FoxPro	None	Character
Visual C++	None	CString object

