NETMANAGE

HTML ActiveX Control

The following Help Topics are available:

For Help on Help, Press F1

Net*Manage*

NetManage develops, markets and supports an integrated set of TCP/IP inter-networking applications and development tools for Microsoft Windows. NetManage software facilitates communication, productivity and the administration of personal computers across dissimilar networking environments. The Company's award-winning product families include Chameleon and ECCO.

The company is located at 10725 North De Anza Blvd. Cupertino, CA 95014, USA

Phone: 408-973-7171 Fax: 408-973-8272.

International phone: +972-4-8550234 Fax +972-4-8550122

General

The HTML Control provides parsing and layout of HTML data, as well as a scrollable view of the selected HTML page. It lets you implement an HTML viewer, with or without automatic network retrieval of HTML documents. The HTML Control can also be used as a non-visual HTML parser to analyze or process HTML documents.

Features

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)
- DocStream interfaces for flexible data transfer
- Events for overriding default processing

Retrieving HTML Data

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

- explicitly. - You can call 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 be activated.
- by selecting an active link and activating the DoRequestDoc event. When you click on an active link, the DoRequestDoc event is activated 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 activating 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 activating 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.

DocStreams

DocStream (DocInput and DocOutput) objects are the mechanism used for data retrieval and submission. You can use the DoRequestDoc, DoRequestEmbedded and DoRequestSubmit events to specify any DocStream for document retrieval or submission. By default, DocStream objects are created internally by the HTML Control for documents with HTTP and File URLs. To provide for other URL types, or specify a DocStream object for any URL type, you can set DocStream properties during event handling.

For more information on Docstream objects, see the DocInput and DocOutput properties, DocInput and DocOutput events, and Common Control Objects.

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 query the attributes and values of the parsed element when DoNewElement is activated. If this event is canceled, parsing will continue but the HTML Control will not store the element.

Using the HTML Control

When using the HTML Control with built-in network document retrieval, or when linking it to other network controls for document retrieval, there should be no added overhead for transfer of data between objects, i.e., there should be no copying of data and notifications should perform as well as ordinary C++ function calls.

No event handling should be necessary to implement a simple Web viewer with browsing and form submission capabilities when using the basic features of the HTML Control. For more powerful use of the control, you can override all built-in document retrieval, browser and form submission behavior.

When using the control as a non-visual parser, there should be no overhead for visual (dormant) aspects of the control.

The HTML Control requires dual (direct call) OLE interfaces. It also uses and is dependent on the DocInput , DocOutput and DocHeader[s] objects.

Each object described is preceded by the required parameter: object. During execution object translates to the name of the OCX. For the HTML OCX object becomes HTML.

Limitations

The following features are not supported by the HTML Control:

- Text selection and clipboard copy
- Automatic external viewer launching
- Proxy server determination and usage
- Built-in FTP retrieval and inline FTP listings
- Basic Authorization and SSL/PCT
- Multipart document submission (file upload)

Naming Conventions

Objects described in the Properties, Methods and Events section are preceded by the required parameter: object. During execution object translates to the name of the relative control or collection name. For HTML actual object name will be:

```
HTMLn
```

where *n* is the number identifier. For example, the first HTML in a form becomes HTML1, the second is HTML2 and so forth.

When using a collection, object becomes the collection name. For example, HTMLAttr collection objects are preceded by HTMLAttr.

Properties

Note: Some common ActiveX properties of the control, such as Name, Index, About Box, and others, may appear in the Object Browser but are not documented here.

HTML properties are:

BackColor

BackImage

BaseURL

DeferRetrieval

DocBackColor

DocForeColor

DocInput

DocLinkColor

DocOutput

DocVisitedColor

ElemNotification

FixedFont

Font

ForeColor

Forms

Heading1Font

Heading2Font

Heading3Font

Heading4Font

Heading5Font

Heading6Font

LayoutDone

LinkColor

ParseDone

Redraw

RequestURL

RetainSource

RetrieveBytesDone

RetrieveBytesTotal

SourceText

Timeout

TotalHeight

TotalWidth

UnderlineLinks

URL

UseDocColors

ViewSource

VisitedColor

BackColor

Description

Defines the default background color.

Syntax

object.**BackColor** [= *color*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

BackColor

Range

0 to 16777215 (&HFFFFFF)

Comments

May be overridden by the DocBackColor property, if such a document color is present and the UseDocColors property is True.

BackImage

Description

URL of an image to be used as the background image of the document.

Syntax

object.**BackImage** [= *String*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

Empty string.

Range

Valid URL.

Comments

May 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

Description

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.

Syntax

object.**BaseURL**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

String.

Default Value

Empty string.

Range

Valid URL.

Comments

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

DeferRetrieval

Description

Indicates whether retrieval of embedded objects should be deferred until explicitly requested.

Syntax

object.**DefRetrieval** [= *Boolean*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

Boolean.

Default Value

False.

Range

True or False.

Comments

The user can set this property to turn inline retrieval of embedded documents off or on. If you are implementing caching, you will normally leave this property set to False so that cached documents are always displayed inline.

DocBackColor

Description

Document background color.

Syntax

object.**DocBackColor** [= *color*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read Only).

Availability

R (Runtime).

Data Type

Long.

Default Value

BackColor.

Range

Valid color.

Comments

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

Description

Document foreground (text) color.

Syntax

object.**DocForeColor** [= *color*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read Only).

Availability

R (Runtime).

Data Type

Long.

Default Value

ForeColor.

Range

Valid color.

Comments

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.

DocInput

Description

Object describing input information for the main document being transferred.

Syntax

object.**DocInput**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

DocInput.

Default Value

N/A.

Range

N/A

Comments

The DocInput object provides a more powerful interface than the basic capabilities of the RequestDoc method. However, you can use the basic functions of the control without knowledge or use of the DocInput object.

DocLinkColor

Description

Document link color.

Syntax

object.**DocLinkColor** [= *color*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only)

Availability

R (Runtime).

Data Type

Long.

Default Value

LinkColor.

Range

Valid color.

Comments

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.

DocOutput

See Also

Description

Object describing output information when submitting form data.

Syntax

object.**DocOutput**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

DocOutput.

Default Value

N/A.

Range

N/A

Comments

The DocOutput object provides a more powerful interface than the basic capabilities of the RequestSubmit method.   However, you can use the basic functions of the control without knowledge or use of the DocInput object.

RequestSubmit
DoRequestSubmit
DocOutputEvent
Common Control Objects

DocVisitedColor

Description

Document visited link color.

Syntax

object.**DocVisitedColor** [= *color*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only)

Availability

R (Runtime).

Data Type

Long.

Default Value

VisitedColor.

Range

Valid color.

Comments

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

See Also

Description

Indicates whether the DoNewElement event should be activated during HTML parsing.

Syntax

object.**ElemNotification** [= *Boolean*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

Boolean.

Default Value

False.

Range

True or False.

Comments

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

DoNewElement

FixedFont

Description

Font for fixed-width text.

Syntax

object.**FixedFont** [=*String*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

Courier New, size 10.

Range

Valid font.

Font

Description

Font for regular text.

Syntax

object.**Font** [=*String*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

Times New Roman, size 12.

Range

Valid font.

ForeColor

Description

Default foreground (text) color.

Syntax

object.**ForeColor** [= *color*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

container ForeColor.

Range

Valid color.

Comments

This property may be overridden by the DocForeColor property if such a document color is present and the UseDocColors property is True.

Forms

See Also

Description

A collection of the forms contained in the HTML page.

Syntax

object.**Forms**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only)

Availability

R (Runtime).

Data Type

HTMLForms.

Default Value

None.

Range

None.

Comments

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

HTMLForms Collection

Heading1Font

Description

Font for heading level 1 text (<H1> elements).

Syntax

object.**Heading1Font** [=*String*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

Times New Roman, size 24, Bold.

Range

Valid font.

Heading2Font

Description

Font for heading level 2 text (<H2> elements).

Syntax

object.**Heading2Font** [=*String*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

Times New Roman, size 18, Bold.

Range

Valid font.

Heading3Font

Description

Font for heading level 3 text (<H3> elements).

Syntax

object.**Heading3Font** [=*String*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

Times New Roman, size 14, Bold.

Range

Valid font.

Heading4Font

Description

Font for heading level 4 text (<H4> elements).

Syntax

object.**Heading4Font** [=*String*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

Times New Roman, size 12, Bold.

Range

Valid font.

Heading5Font

Description

Font for heading level 5 text (<H5> elements).

Syntax

object.**Heading5Font** [=*String*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

Times New Roman, size 10, Bold.

Range

Valid font.

Heading6Font

Description

Font for heading level 6 text (<H6> elements).

Syntax

object.**Heading6Font** [=*String*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

Times New Roman, size 8, Bold.

Range

Valid font.

LayoutDone

Description

Indicates whether the layout phase is complete.

Syntax

object.**LayoutDone**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only)

Availability

R (Runtime).

Data Type

Boolean.

Default Value

False.

Range

True or False.

Comments

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

Description

Default link color.

Syntax

object.**LinkColor** [= *color*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

blue (0,0,255)

Range

Valid color.

Comments

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

ParseDone

Description

Indicates whether the parsing phase is complete.

Syntax

object.**ParseDone**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only)

Availability

R (Runtime).

Data Type

Boolean.

Default Value

False.

Range

True or False.

Comments

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

Redraw

Description

Indicates whether drawing should occur as data changes or the window is scrolled.

Syntax

object.**Redraw** [= *Boolean*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

Boolean.

Default Value

True.

Range

True or False.

Comments

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

Description

URL string identifying the new document requested.

Syntax

*object***.RequestURL**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read Only).

Availability

R (Runtime).

Data Type

String.

Default Value

Empty String.

Range

Valid URL.

Comments

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

RetainSource

Description

Indicates whether source text should be retained and available via the SourceText property.

Syntax

object.**RetainSource** [= *Boolean*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

Boolean.

Default Value

True.

Range

True or False.

Comments

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

RetrieveBytesDone

Description

Completed byte size of the objects being retrieved.   This property is zero if no retrieval is in progress.

Syntax

*object***.RetrieveBytesDone**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only)

Availability

R (Runtime).

Data Type

Long.

Default Value

Zero.

Range

>=zero.

RetrieveBytesTotal

Description

Total byte size of the objects to be retrieved, including embedded objects and the document itself.

Syntax

object.**RetrieveBytesTotal**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only)

Availability

R (Runtime).

Data Type

Long.

Default Value

Zero.

Range

>=zero.

Comments

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

Description

Contains the source text of the main document.

Syntax

object.**SourceText**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only)

Availability

R (Runtime).

Data Type

String.

Default Value

None.

Range

None.

Comments

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

Timeout

Description

Time-out interval (in seconds) for initiating the request for documents. The Timeout event is activated if no data is received within timeout.

Syntax

object.**Timeout** [= *Long*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (design) and R (Runtime).

Data Type

Long.

Default Value

30 seconds.

Range

N/A.

Comments

Although the Timeout value applies to all document retrieval, the Timeout event is activated only for the main document, not for embedded documents. *Event* is an integer value that determines the type of Timeout event that will be enabled. Constants defined for enum types for events are:

Value	Meaning
prcConnectTimeout = 0	Timeout for connect. If connection is not established within the timeout period, the Timeout event will be activated.
prcReceiveTimeout = 1	Timeout for receiving data. If no data arrives within the timeout period, the Timeout event will be activated.
prcUserTimeout= 65	Timeout for user defined event. Use prcUserTimeout + [Integer] range for custom timeout events.

TotalHeight

Description

Total height of the document in pixels.

Syntax

object.**TotalHeight**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only)

Availability

R (Runtime).

Data Type

Long.

Default Value

Zero.

Range

>=zero.

Comments

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

Description

Total width of the document in pixels.

Syntax

*object***.Totalwidth**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only)

Availability

R (Runtime).

Data Type

Long.

Default Value

Zero.

Range

>=zero.

Comments

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

Description

Indicates whether links should be underlined.

Syntax

object.**UnderlineLinks** [= *Boolean*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

Boolean.

Default Value

True.

Range

True or False.

URL

Description

URL string identifying the current main document.

Syntax

object.**URL**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read Only).

Availability

R (Runtime).

Data Type

String.

Default Value

Empty String.

Range

Valid URL.

Comments

This property is set by the control from the URLRequest property when document retrieval has successfully started and the BeginRetrieval event is activated.

UseDocColors

Description

Indicates whether document colors should be used when present.

Syntax

object.**UseDocColors** [= *Boolean*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

Boolean.

Default Value

True.

Range

True or False.

Comments

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 will be used to display active links; otherwise, the LinkColor property value will be used.

ViewSource

Description

Indicates whether the control should display HTML source as plain text.

Syntax

object.**ViewSource** [= *Boolean*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

Boolean.

Default Value

False.

Range

True or False.

Comments

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 will be initiated to obtain the source text for viewing.

VisitedColor

Description

Default visited link color.

Syntax

object.**VisitedColor** [= *color*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

W (Read/Write).

Availability

D (Design) and R (Runtime).

Data Type

String.

Default Value

purple (255,0,255)

Range

Valid color.

Comments

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

Methods

The HTML methods are:

Cancel

RequestAllEmbedded

RequestDoc

Cancel

Description

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

object.**Cancel** [*Message*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

Message

Optional. Message to be appended to the HTML page.
Data Type: String
Param: IN
Default Value: None

Comments

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

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

HTML tags are also allowed in the Message.

RequestAllEmbedded

Description

Requests retrieval of all embedded documents via the DoRequestEmbedded event.

Return Value

Void.

Syntax

object.**RequestAllEmbedded**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

None.

Comments

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

RequestDoc

Description

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

Return Value

Void.

Syntax

object.**RequestDoc** *URL*

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

URL

Identifies the new main document to be retrieved.
Data Type: String
Param: IN
Default Value: http:

Comments

When RequestDoc is called, the DoRequestDoc event is activated to determine the DocStream to be used for retrieval. The RequestURL property will then be set to the URL parameter specified. The URL property will not be updated until retrieval is successfully underway and the BeginRetrieval event is activated.

Events

Following is a list of the events activated by the HTML Control. Each description includes the syntax, related parameters, their data type, default value, and whether the parameter is used for input or output (IN or OUT).

BeginRetrieval

Click

DblClick

DocInput

DocOutput

DoNewElement

DoRequestDoc

DoRequestEmbedded

DoRequestSubmit

EndRetrieval

GotFocus

KeyDown

KeyPress

KeyUp

LayoutComplete

LostFocus

MouseDown

MouseMove

MouseUp

ParseComplete

TimeOut

UpdateRetrieval

BeginRetrieval

Description

This event is activated when document retrieval begins.

Syntax

*object*_**BeginRetrieval**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

None.

Comments

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

Click

Description

This event is activated when the user presses and then releases the mouse button over an object.

Syntax

*object*_**Click**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

None.

DblClick

Description

This event is activated when the user presses and releases the mouse button twice over an object.

Syntax

*object*_**DblClick**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

None.

DocInput

See Also

Description

A DocInput related event that indicates the input data has been transferred.

Syntax

*object*_**DocInput** (*DocInput* **As DocInput**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

DocInput

Object describing document input data for the current transfer.
Data Type: DocInput
Param: IN
Default Value: N/A

Comments

The DocInput event can be used in its basic form for notification of transfer progress, (for example, for updating a progress bar). The DocInput.BytesTotal, DocInput.BytesTransferred and DocInput.State properties can be examined to determine the current status of the transfer. This event can be ignored if no progress information is needed.

To gain more power from this control, you can also use the DocInput event for data streaming.

Common Control Objects

DocOutput

See Also

Description

A DocOutput related event indicating that output data has been transferred.

Syntax

*object*_**DocOutput** (*DocOutput* **As DocOutput**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

DocOutput

Object describing document output data for the current transfer.
Data Type: DocOutput
Param: IN
Default Value: N/A

Comments

The DocOutput event can be used in its basic form to notify the user of transfer progress, (for example, for updating a progress bar). The DocOutput.BytesTotal, DocOutput.BytesTransferred and DocOutput.State properties can be examined to determine the current status of the transfer. This event can be ignored if no progress information is needed.

To gain more power from this control, you can also use the DocOutput event for data streaming.

Common Control Objects

DoNewElement

Description

The event is activated during HTML parsing when a new element is added.

Syntax

*object*_**DoNewElement** (*ElemType* **As String**, *EndTag* **As Boolean**, *Attrs* **As HTMLAttrs**, *Text* **as String**, *EnableDefault* **As Boolean**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

Elemtype

Element type for tags; empty string for character data.
Data Type: String
Param: IN
Default Value: None.

EndTag

True if an end tag; otherwise False.
Data Type: Boolean
Param: IN
Default Value: None.

Attrs

Collection of tag attributes, described in HTMLAttrs Collection Properties
Data Type: HTMLAttrs
Param: IN
Default Value: None.

Text

Character data; empty for tags.
Data Type: String
Param: IN
Default Value: None.

EnableDefault

Overrides default processing. True indicates default processing, False means override defaults. If EnableDefault is set to false, the HTML Control does not store data for this element, but continues parsing.
Data Type: Boolean
Param: IN/OUT
Default Value: True.

Comments

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.

DoRequestDoc

Description

The event is activated when the user chooses a link to a different URL or when the RequestDoc method is called.

Syntax

*object*_**DoRequestDoc** (*URL* **As String**, *Element* **As HTMLElement**, *DocInput* **As DocInput**, *EnableDefault* **As Boolean**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

URL

Identifies the requested document
Data Type: String
Param: IN
Default Value: None.

Element

Currently unused, but in the future it will identify the anchor element of the link selected by the user.
Data Type: HTMLElement
Param: IN
Default Value: None.

DocInput

Causes the control to accept input from another source.
Data Type: DocInput
Param: IN
Default Value: None.

EnableDefault

Overrides default processing. True indicates default processing, False means cancel default processing.
Data Type: Boolean
Param: IN/OUT
Default Value: True.

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

Comments

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

DoRequestEmbedded

See Also

Description

The event is activated when an embedded document, such as an image is to be retrieved for inline display

Syntax

*object*_**DoRequestEmbedded** (*URL* **As String**, *Element* **As HTMLElement**, *DocInput* **As DocInput**, *EnableDefault* **As Boolean**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

URL

Identifies the requested document
Data Type: String
Param: IN
Default Value: None.

Element

Currently unused, but in the future it will identify the HTML element of the embedded document.
Data Type: HTMLElement
Param: IN
Default Value: None.

DocInput

Causes the control to accept input from another source.
Data Type: DocInput
Param: IN
Default Value: None.

EnableDefault

Overrides default processing. True indicates default processing, False means cancel the request.
Data Type: Boolean
Param: IN/OUT
Default Value: True.

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.

DocInput Property
DocInputEvent
Common Control Objects

DoRequestSubmit

Description

The event is activated when the user selects form submission, or when the RequestSubmit method of the Form is called.

Syntax

*object*_**DoRequestSubmit** (*URL* **As String**, *Form* **As HTMLForm**, *DocOutput* **As DocOutput**, *EnableDefault* **As Boolean**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

URL

Identifies the action URL for that form, and includes the search string for GET form methods as described in Comments.
Data Type: String
Param: IN
Default Value: None.

Form

Identifies the form being submitted, and is an item in the Forms collection
Data Type: HTMLForm
Param: IN
Default Value: None.

DocOutput

Causes output to another target.
Data Type: DocOutput
Param: IN

EnableDefault

Overrides default processing. True indicates default processing, False means override defaults. To cancel the submission request, set the EnableDefault parameter to False. If the event is not canceled, the RequestURL property will be set by the control. The URL property will not be updated until retrieval is successfully underway and the BeginRetrieval event is activated.
Data Type: Boolean
Param: IN/OUT
Default Value: True.

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 using a different source and/or target during event handling, you may 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.

Comments

Currently, the form contents for submission always consist of URL-encoded field values contained in the Form.URLEncodedBody property. In the future, multipart content data may 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

Description

The event is activated when document retrieval, including embedded documents to be displayed inline, is complete.

Syntax

*object*_**EndRetrieval**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

None.

Comments

A progress bar could be terminated at this time.

GotFocus

Description

The event is activated when an object receives the focus, either by user action, such as tabbing to or clicking the object, or by changing the focus in the code using the SetFocus method.

Syntax

*object*_**GotFocus**

Parameters

None.

KeyDown

Description

The event is activated when the user presses a key while an object has the focus. All arguments of this event are standard.

Syntax

*object*_**KeyDown** (*KeyCode* **As Integer**, *Shift* **As Integer**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

KeyCode

Indicates the key being pressed.
Data Type: Integer
Param: IN
Default Value: None.

Shift

Indicates whether a Shift, Ctrl, and/or Alt key was also pressed.
Data Type: Integer
Param: IN
Default Value: None.

KeyPress

Description

The event is activated when the user presses and releases an ANSI key. All arguments of this event are standard.

Syntax

*object*_**KeyPress** (*KeyAsii* **As Integer**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

KeyAsii

Indicates the ANSI key being pressed.
Data Type: Integer
Param: IN
Default Value: None.

KeyUp

Description

The event is activated when the user releases a key while an object has the focus.   All arguments of this event are standard.

Syntax

*object*_**KeyUp** (*KeyCode* **As Integer**, *Shift* **As Integer**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

KeyCode

Indicates the key being released.
Data Type: Integer
Param: IN
Default Value: None.

Shift

Indicates whether a Shift, Ctrl, and/or Alt key was also pressed.
Data Type: Integer
Param: IN
Default Value: None.

LayoutComplete

Description

The event is activated when layout of the HTML document is complete.

Syntax

*object*_**LayoutComplete**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

None.

Comments

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

LostFocus

Description

The event is activated when an object loses the focus, either by user action, such as tabbing to or clicking the object, or by changing the focus in the code using the SetFocus method.

Syntax

*object*_**LostFocus**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

None.

MouseDown

Description

The event is activated when the user presses a mouse button.   All arguments of this event are standard.

Syntax

*object*_**MouseDown** (*Button* **As Integer**, *Shift* **As Integer,** *X* **As Float,** *Y* **As Float**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

Button

Indicates which mouse button was pressed.
Data Type: Integer
Param: IN
Default Value: None.

Shift

Indicates whether a Shift, Ctrl, and/or Alt key was also pressed.
Data Type: Integer
Param: IN
Default Value: None.

X

Indicates the X axis of the pointer when the mouse button was pressed.
Data Type: Integer
Param: IN
Default Value: None.

Y

Indicates the Y axis of the pointer when the mouse button was pressed.
Data Type: Integer
Param: IN
Default Value: None.

MouseMove

Description

The event is activated the user moves the mouse.   All arguments of this event are standard.

Syntax

*object*_**MouseMove** (*Button* **As Integer**, *Shift* **As Integer,** *X* **As Float,** *Y* **As Float**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

Button

Indicates the mouse button being moved.
Data Type: Integer
Param: IN
Default Value: None.

Shift

Indicates whether a Shift, Ctrl, and/or Alt key was also pressed.
Data Type: Integer
Param: IN
Default Value: None.

X

Indicates the X axis of the pointer when the mouse button was pressed.
Data Type: Integer
Param: IN
Default Value: None.

Y

Indicates the Y axis of the pointer when the mouse button was pressed.
Data Type: Integer
Param: IN
Default Value: None.

MouseUp

Description

The event is activated when the user releases a mouse button. All arguments of this event are standard.

Syntax

*object*_**MouseUp** (*Button* **As Integer**, *Shift* **As Integer,** *X* **As Float,** *Y* **As Float**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

Button

Indicates the mouse button being released.
Data Type: Integer
Param: IN
Default Value: None.

Shift

Indicates whether a Shift, Ctrl, and/or Alt key was also released.
Data Type: Integer
Param: IN
Default Value: None.

X

Indicates the X axis of the pointer when the mouse button was pressed.
Data Type: Integer
Param: IN
Default Value: None.

Y

Indicates the Y axis of the pointer when the mouse button was pressed.
Data Type: Integer
Param: IN
Default Value: None.

ParseComplete

Description

The event is activated when parsing of the HTML document is complete.

Syntax

*object*_**ParseComplete**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

None.

Comments

Layout and embedded document retrieval may not be complete.

TimeOut

Description

The event is activated after no data has been received within the time specified in the TimeOut property.

Syntax

*object*_**TimeOut**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

None.

Comments

Although the Timeout value applies to all document retrieval, the Timeout event is activated only for the main document, not for embedded documents. *Event* is an integer value that determines the type of Timeout event that will be enabled. Constants defined for enum types for events are:

Value	Meaning
prcConnectTimeout = 0	Timeout for connect. If connection is not established within the timeout period, the Timeout event will be activated.
prcReceiveTimeout = 1	Timeout for receiving data. If no data arrives within the timeout period, the Timeout event will be activated.
prcUserTimeout= 65	Timeout for user defined event. Use prcUserTimeout + [Integer] range for custom timeout events.

UpdateRetrieval

Description

The event is activated periodically as the document and embedded objects are retrieved.

Syntax

*object*_**UpdateRetrieval**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

None.

Comments

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

HTMLAttrs Collection

An HTMLAttrs object is a collection containing HTMLAttr items.   An HTMLAttrs collection is passed as an argument when the DoNewElement event is activated.

HTMLAttrs Collection Properties

HTMLAttrs Collection Methods

HTMLAttr Item

HTMLAttrs Collection Properties

The HTMLAttrs collection consists of the Count property.

Count

Description

The number of attributes in the collection.

Syntax

object.**Count**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

Long.

Default Value

None.

Range

>=zero.

HTMLAttrs Collection Methods

The HTMLAttrs collection consists of the Item method.

Item

Description

Returns an item from the collection. The Item method is the default method for a collection.

Syntax

object.**Item** (*Index*)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

Index

Identifies the item in the collection. May be either an integer or a string. Integer indices identify an item by its zero-based index. String indices identify an item by its Name property.
Data Type: Variant
Param: IN
Default Value: None.

HTMLAttr Item

An HTMLAttr object is an item in an HTMLAttrs collection.   HTMLAttr items are used for specifying the attribute names and values of an HTML element.

HTMLAttr Item Properties

HTMLAttr Item Properties

The HTMLAttr Items supports these properties.

Name

Value

Name

Description

The attribute name.   This string is never empty, and may be uppercase, lowercase or mixed case.

Syntax

*object***.Item(***"Name"*)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

String.

Default Value

None.

Range

See RFC 1866 for attribute name syntax.

Value

Description

The attribute value. This string may be empty. If not empty, the string is unescaped (decoded).

Syntax

object.**Item**(*"Value"*)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

String.

Default Value

None.

Range

Any string.

HTMLForms Collection

An HTMLForms object is a collection containing HTMLForm items.   The Forms property of the HTML Control is an HTMLForms collection.

HTMLForms Collection Properties

HTMLForms Collection Methods

HTMLForm Item

HTMLForms Collection Properties

The HTMLForms collection supports the Count property.

Count

Description

The number of forms in the collection.

Syntax

object.**Count**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

Long.

Default Value

None.

Range

>=zero.

HTMLForms Collection Methods

The HTMLForms collection supports Item method.

Item

Description

Returns an item from the collection.   The Item method is the default method for a collection.

Syntax

*object***.Item** (*Index*)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

Index

> Identifies the item in the collection.   Must be either an integer; identifies an item by its zero-based index.
> Data Type: Variant
> Param: IN
> Default Value: None.

HTMLForm Item

An HTMLForm object is an item in an HTMLForms collection.   HTMLForm items are used for submitting documents using HTTP.

HTMLForm Item Properties

HTMLForm Item Methods

HTMLForm Item Properties

The HTMLForms Item supports these properties.

Method

URL

URLEncodedBody

Method

Description

The HTTP submission method for the form.

Syntax

*object***.Item.Method**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

String.

Default Value

None.

Range

Get or Post.

URL

Description

The action URL for the form.

Syntax

*object***.Item.URL**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

String.

Default Value

None.

Range

Valid URL.

URLEncodedBody

Description

The URL-encoded body text, representing the values of all form fields used for HTTP submission.

Syntax

*object***.Item.URLEncodedBody**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

String.

Default Value

None.

Range

URL-encoded string.

HTMLForm Item Methods

The HTMLForms Item supports the RequestSubmit method.

RequestSubmit

Description

Requests submission of a form.

Return Value

Void.

Syntax

object.**Item.RequestSubmit**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

Parameters

None.

Comments

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

ActiveX

ActiveX is a trademark of Microsoft Corporation.

HTML Errors

For this release there are no HTML error codes.