



HTTP Client *ActiveX* Control

The following Help Topics are available:

[General](#)

[Naming Conventions](#)

[Properties](#)

[Methods](#)

[Events](#)

[Error Codes](#)

[Localization](#)

For Help on Help, Press F1

NetManage

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 HTTP (Hypertext Transport Protocol) Control implements the HTTP Protocol Client based on the HTTP specification. This control lets you directly retrieve HTTP documents if no browsing or image processing is necessary.

It can be used by developers who implement HTML browsers or other services that need access to HTTP. For example, the HTML Control internally instantiates this object and uses it for HTTP transactions.

The HTTP Control uses a number of methods to retrieve or send (post) a document. It can retrieve MIME information about the document from the Headers collection property.

There should be no speed overhead and response delay other than the one given by the network. This control uses and is dependent on the DocStream objects (DocInput and DocOutput).

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 control.

Properties

Properties set the attributes for HTTP behavior.

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.

Following is a list of all properties supported by the HTTP Client Control.

[Busy](#)

[DocInput](#)

[DocOutput](#)

[Document](#)

[EnableTimer](#)

[Errors](#)

[Method](#)

[NotificationMode](#)

[ProtocolState](#)

[ProtocolStateString](#)

[RemoteHost](#)

[RemotePort](#)

[ReplyCode](#)

[ReplyString](#)

[State](#)

[StateString](#)

[Timeout](#)

[URL](#)

Busy

Description

Indicates a command is in progress.

Syntax

object.**Busy**

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

N/A.

Range

True or False

DocInput

See Also

Description

Object describing input information for the 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 SendDoc method. However, you can use the basic functions of the control without knowledge or use of the DocInput object.

Properties of the DocInput object may be set before calling the SendDoc method or they may be passed as arguments to this method. The DocInput object is also used for conveying information about the progress of the document transfer and for data linking and streaming.

DocInput Event

Common Control Objects

DocOutput

See Also

Description

Object describing output information for the document being transferred.

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 GetDoc method. However, you can use the basic functions of the control without knowledge or use of the DocInput object.

Properties of the DocOutput object may be set before calling the GetDoc method or they may be passed as arguments to this method. The DocOutput object is also used for conveying information about the progress of the document transfer, and for data linking and streaming.

[DocOutput](#) Event

[Common Control Objects](#)

Document

Description

Identifies the target document. The Document property can be used with RemoteHost to identify the URL. It can also be used instead of URL.

Syntax

object.Document [= *String*]

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

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

String.

Default Value

Empty String.

Range

N/A.

EnableTimer

Description

Enable timer for the specified event. The event is specified by entering:

```
EnableTimer(short event)
```

Syntax

object.EnableTimer (event) [= Boolean]

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

Permission

W (Write Only).

Note: This is the only control property that is Write only.

Availability

R (Runtime)

Data Type

Boolean.

Default Value

False. (The timer for this event will not be enabled.)

Range

True or False

Comments

Event is an integer value that determines the type of Timeout event that will be enabled. Constants defined for enum types for events are:

<u>Value</u>	<u>Meaning</u>
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.

Errors

Description

A collection of errors that can be accessed for details about the last error that occurred. This collection should be used within an Error event if information passed through the Error event is not sufficient. For more details, see [icErrors](#).

Syntax

object.Errors

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

icErrors.

Default Value

N/A.

Range

N/A

Method

Description

Method used to retrieve or post (send) the document.

Syntax

object.Method [= *Integer*]

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

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

Integer.

Default Value

1 (prcGet)

Range

1-4. The possible values are:

Value	Meaning
prcGet = 1	Get method request the whole document.
prcHead = 2	Head method requests only the headers of a document.
prcPost = 3	Post method posts the whole document to the server as a sub-ordinate of the document specified by the URL.
prcPut = 4	Put method puts the whole document to the server. The document replaces an existing document specified by the URL.

NotificationMode

Description

Determines when notification is issued for incoming data. Notification can also be suspended.

Syntax

object.NotificationMode [= *Integer*]

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

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

Integer.

Default Value

1.

Range

0-maximum unsigned long. At present , the values are:

Constant	Meaning
0	COMPLETE: notification is provided when there is a complete response.
1	CONTINUOUS: an event is repeatedly activated when new data arrives from the connection.

ProtocolState

Description

This property specifies the current state of the protocol.

Syntax

object.ProtocolState

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

Integer.

Default Value

prcBase.

Range

0-1. Constants defined for the enum types of ProtocolState property are:

Value	Meaning
prcBase = 0	Base state before connection to server is established.
prcTranferring = 1	Data is being transferred.

ProtocolStateString

Description

String representation of ProtocolState.

Syntax

object.**ProtocolStateString**

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

BASE.

Range

N/A.

RemoteHost

Description

The remote machine to connect to if the remoteHost parameter in the Connect method is missing. You can either provide a host name or an IP address string in dotted format. For example, 127.0.0.1.

Note: This is the default property of the control.

Syntax

object.RemoteHost [= *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).

Data Type

String.

Default Value

Empty.

Range

N/A.

RemotePort

Description

The remote port number to which to connect.

Syntax

object.RemotePort [= *Long*]

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

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

Long.

Default Value

80.

Range

1-65535.

ReplyCode

Description

The value of the reply code is a protocol specific number that determines the result of the last request, as returned in the ReplyString property.

Syntax

object.**ReplyCode**

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

0.

Range

0- (still undefined)

ReplyString

Description

Lists the last reply string sent by the HTTP Server to the client as a result of a request.

Syntax

object.**ReplyString**

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.

Range

N/A.

State

Description

This property specifies the connection state of the control.

Syntax

object.State

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

Integer.

Default Value

prcDisconnected.

Range

1-6. Constants defined for enum types of State property are:

Value	Meaning
prcConnecting = 1	Connecting. Connect has been requested, waiting for connect acknowledge.
prcResolvingHost = 2	Resolving Host. Occurs when RemoteHost is in name format rather than dot-delimited IP format.
prcHostResolved = 3	Resolved the host. Occurs only if ResolvingHost state has been entered previously.
prcConnected = 4	Connection established.
prcDisconnecting = 5	Connection closed. Disconnect has been initiated.
prcDisconnected = 6	Initial state when protocol object is instantiated, before Connect has been initiated, after a Connect attempt failed or after Disconnect performed.

StateString

Description

A string representation of State.

Syntax

object.StateString

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

"Disconnected".

Range

N/A

Timeout

Description

Timeout value for the specified event. The event is specified by entering:

`Timeout (short event)`

Syntax

`object.Timeout (event) [= Long]`

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

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

Long.

Default Value

0.

Range

0-maximum unsigned long. Constants defined for enum types for events are:

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

URL

Description

URL string identifying the current document being transferred. URL format is:

`HTTP://host:port/documentnameandpath`

Syntax

`object.URL [= String]`

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

Permission

W (Read/Write).

Availability

R (Runtime).

Data Type

String.

Default Value

Empty string.

Range

Valid URL.

Comments

URL may be set before calling the GetDoc or SendDoc method of the control, or it may be passed as an argument to these methods. If it is passed as an argument, the URL property will be set to the argument value.

In the HTTP Control, the URL property identifies an HTTP request of any kind. The URL type (first part up to the colon) may be omitted. In this case, it will default to the correct type for this control. For example, the `http:` string may be omitted when using the HTTP Control.

Methods

The methods performed by the HTTP Client Control are:

[Cancel](#)

[Connect](#)

[GetDoc](#)

[PerformRequest](#)

[SendDoc](#)

Cancel

Description

Cancels a pending request.

Return Value

Void.

Syntax

object.**Cancel**

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

Parameters

None.

Connect

Description

Initiates a Connect request. The control calls the StateChanged event if a connection is established.

Return Value

Void.

Syntax

object.**Connect** [*RemoteHost*,] [*RemotePort*]

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

Parameters

RemoteHost

Optional. Remote host to which to connect. If this parameter is missing, the control connects to the host defined in the RemoteHost property.

Data Type: String

Param: IN

Default Value: N/A

RemotePort

Optional. Remote port to which to connect. If this parameter is missing, the control connects to the port defined in the RemotePort property.

Data Type: Long

Param: IN

Default Value: N/A

Comments

Optional arguments to this method override the values from corresponding RemoteHost and RemotePort properties. The values of the properties will not change. If no argument is given, the values from the properties will be used to establish the connection.

GetDoc

See Also

Description

A DocOutput related method that requests retrieval of a document identified by a URL.

Return Value

Void.

Syntax

object.**GetDoc** [*URL*,] [*Headers*,] [*OutputFile*]

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

Parameters

URL

Optional. The URL identifying the remote document to be retrieved.

Data Type: String

Param: IN

Default Value: DocInput.URL

Headers

Optional. Headers used for requesting the document. This argument only applies to protocols where request headers can be specified (for example, HTTP).

Data Type: DocHeaders

Param: IN

Default Value: DocInput.Headers

OutputFile

Optional. A local file to which the retrieved document will be written.

Data Type: String

Param: IN

Default Value: DocOutput.Filename

Comments

The GetDoc method permits retrieving a document from the server.

The URL and (for some controls) Headers are used as inputs specifying which document is to be retrieved. The OutputFile argument indicates where the retrieved document should be written locally.

The URL type (first part up to the colon) may be omitted and will default to the correct type for this control. For example, when using the HTTP Control, the "http:" string may be omitted.

For basic use of this control, arguments should be passed to GetDoc to describe the document transfer. For more powerful use of this control, the DocInput and DocOutput objects can be used in conjunction with the DocInput and DocOutput events. The arguments of GetDoc correspond to properties in the DocInput and DocOutput objects of this control. DocInput and DocOutput properties can be set before calling GetDoc to avoid passing arguments. The DocInput and DocOutput events can also be used for transferring data using streaming rather than local files.

DocInputProperties

DocOutput Properties

DocInput, DocOutput Events

Common Control Objects

PerformRequest

Description

Initiates a request method to retrieve a document. If no parameters are specified, properties Document, HostName, RemotePort and Method are used for the retrieval. This method is similar to GetDoc, except it uses a different set of parameters.

Return Value

Void.

Syntax

object.**PerformRequest** [*HostName*,] [*Document*,] [*Method*,] [*RemotePort*]

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

Parameters

HostName

Optional. Host name to use for the request. This value overrides the RemoteHost property.

Data Type: String

Param: IN

Default Value: None

Document

Optional. Document to be retrieved. This value overrides the Document property.

Data Type: String

Param: IN

Default Value: None

Method

Optional. Method to be used for the retrieval. This value overrides the Method property.

Data Type: Integer

Param: IN

Default Value: None

RemotePort

Optional. Remote port number to use when connecting. This value overrides the RemotePort property.

Data Type: Long

Param: IN

Default Value: None

Comments

The [DocOutput](#) event is activated when the data is available.

SendDoc

See Also

Description

A DocInput related method that requests sending a document identified by a URL.

Return Value

Void.

Syntax

object.SendDoc [*URL*,] [*Headers*,] [*InputData*,] [*InputFile*,] [*OutputFile*]

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

Parameters

URL

Optional. The URL identifying the remote document to be sent. If specified, the URL property will be set to this value.

Data Type: String

Param: IN

Default Value: DocInput.URL

Headers

Optional. Headers used for sending the document. This argument only applies to protocols where document headers can be sent (for example, SMTP and HTTP).

Data Type: DocHeaders

Param: IN

Default Value: DocInput.Headers

InputData

Optional. A data buffer containing the document to be sent.

Data Type: VARIANT

Param: IN

Default Value: DocInput.SetData

InputFile

Optional. A local file containing the document to be sent.

Data Type: String

Param: IN

Default Value: DocInput.Filename

OutputFile

Optional. A local file to which a reply document is written. This argument only applies for protocols that return a reply document (for example, HTTP).

Data Type: String

Param: IN

Default Value: DocOutput.Filename

Comments

The SendDoc method permits sending (posting or putting) a file to the server.

The URL and (for some controls) Headers are used as inputs describing the document to be sent.

The InputData and InputFile arguments (only one can be specified) contain the document to be sent.

For controls such as HTTP that return a reply document, the OutputFile argument indicates where the reply document should be written locally.

The URL type (first part up to the colon) may be omitted and will default to the correct type for this control. For example, when using the HTTP Control, the "http:" string may be omitted .

For basic use of this control, arguments should be passed to SendDoc to describe the document transfer. For more powerful use of this control, the DocInput and DocOutput objects can be used in conjunction with the DocInput and DocOutput events. The arguments of SendDoc correspond to properties in the DocInput and DocOutput objects of this control. DocInput and DocOutput properties can be set before calling SendDoc to avoid passing arguments. The DocInput and DocOutput events can also be used for transferring data using streaming rather than local files.

DocInputProperties

DocOutput Properties

DocInput, DocOutput Events

Common Control Objects

Events

Events are used for HTTP client notification. They indicate that an action has been requested and processed. Any errors which occur during command processing result in the Error event being called with appropriate error codes. Error codes, state changes, and protocol return values are usually checked during event processing.

Each event description includes the syntax, related parameters, their data type, default value, and whether the parameter is used for input or output (IN or OUT).

[Busy](#)

[Cancel](#)

[DocInput](#)

[DocOutput](#)

[Error](#)

[ProtocolStateChanged](#)

[StateChanged](#)

[TimeOut](#)

Busy

Description

This event is activated when a command is in progress or when a command has completed.

Syntax

*object_***Busy** (*Busy As Boolean*)

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

Parameters

Busy

Indicates whether or not a command is in progress.

Data Type: Boolean. If the argument is True, a command is in progress.

Cancel

Description

This event is activated after a cancellation request has been completed and satisfied. After this event the object's state changes to idle.

Syntax

*object*_**Cancel**

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

Error

Description

This event is activated when an error occurs in background processing (for example, failed to connect or failed to send or receive in the background).

Syntax

object_Error (**ErrCode As Integer**, **Description As String**, **SCode As Long**, **Source As String**, **HelpFile As String**, **HelpContext As Long**, **CancelDisplay As Boolean**)

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

Parameters

ErrCode

The short error code.

Description

String containing error information.

sCode

The long Scode.

Source

Error source.

HelpFile

Help file name.

HelpContext

Help file context.

CancelDisplay

Indicates whether to cancel the display. The default is TRUE (no display of the default error message box). If you do want to use the default message box, set CancelDisplay to FALSE.

ProtocolStateChanged

Description

This event is activated whenever the protocol state changes.

Syntax

*object*_ProtocolStateChanged (State As Integer)

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

Parameters

Refer to the [ProtocolState](#) property and [ProtocolStateString](#) for possible values of the state parameter.

StateChanged

Description

This event is activated whenever the state of the transport state changes.

Syntax

object_StateChanged (*State As Integer*)

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

Parameters

Refer to the [State](#) property for possible values of the state parameter.

TimeOut

See Also

Description

This event is activated when the timer for the specified event expires. See Timeout property for pre-defined events.

Syntax

object_TimeOut (ByVal *Event As Integer*, *Continue As Boolean*)

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

Parameters

Event

Defines the event to which the time interval applies.

Data Type: Integer

Continue

Determines if the timer is active or not. Set *Continue* to TRUE to keep the timer active.

Data Type: Boolean

Default Value: False

Comments

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.

TimeOut Property

Localization

Error strings for this control are localized.

ActiveX

ActiveX is a trademark of Microsoft Corporation.

HTTP Errors

For this release there are no HTTP error codes.

