



Getting Started With the CIHTTP Control

Related Topics

This topic introduces the Hypertext Transfer Protocol (HTTP) and the Crescent Internet Toolpak HTTP control (CIHTTP). You can access the following topics:

- [What HTTP is](#)
- [What the CIHTTP control is](#)
- [What CIHTTP Programming tasks are](#)

The programming tasks topics identify each task that CIHTTP supports; it lists the properties to set and the methods to call, then it illustrates the task with a code sample.



What is HTTP?

Related Topics

The Hypertext Transfer Protocol lets you transfer data in HTML format. HTML is the *HyperText Markup Language* that is used in World Wide Web (WWW) documents. HTML lets you define HyperText links. A *hypertext link* is a type of formatting that lets you link a document to another document or lets you link items within a document. Hypertext links can be used to link documents from any WWW site in the world. They allow users to proceed from one Web site to the next. This process is sometimes called surfing. The links are embedded in the Web documents. Web documents can contain text, images, and links. You use a *browser* to view an HTML page.



What Is the CIHTTP Control?

Related Topics

The Crescent Internet Hypertext Transfer Protocol control enables Visual Basic programmers to write customized World Wide Web (WWW) client applications. Client applications can retrieve text or binary data from a Web site, retrieve data about a page at the Web site or post text to a Web site from within an OLE container. The control can also be used to enhance the functionality of Web browsers.

CIHTTP has a simple programming interface in which you set property values and call methods to perform the desired tasks. When you call a method (which generally does not take arguments), the control takes the data it needs directly from the property sheet and submits it along with the request to the HTTP server. When the server returns a response, the control parses the data and updates, or refreshes, the properties as required so you can simply read the data from a property value or a ListBox object.



Programming Tasks

Related Topics

The CIHTTP control lets you build HTTP client applications. Each client application must contain code that connects to an HTTP server before it can issue any requests to the server. The connection requirements include;

1. Setting the HostName or HostAddress property and the HTTPPort property.
2. Calling the ConnectToHTTPServer method.

The code looks like this;

```
CIHTTP1.HostName = www.myhost.com
CIHTTP1.HTTPPORT = 80
nResult% = CIHTTP.ConnectToHTTPServer
```

When the connection attempt succeeds, *nResult%* is an integer that represents the socket number opened by the ConnectToHTTPServer method, and the HTTPServerConnection event fires. When the connection attempt fails, *nResult%* is 0 and the WSAError event fires.

By nature, HTTP is transaction oriented, that is, you submit a request to an HTTP server, the server fulfills the request which completes the transaction. When a transaction completes, the server ends the connection. Thus, you can verify whether a transaction is complete by adding code to the FileClosed event module. You know that an error occurred when you get no data in response to a GET or HEAD method. When the POST method fails, the server returns an error code to the PacketReceived event.

Once connected, you can retrieve a web page, post data to a web site



Retrieve a Web Page

Related Topics

This topic illustrates how you can use CIHTTP to retrieve a page from a Web site and store it to a file.

1. Set these properties, then call the ConnectToHTTPServer.

HostName (or HostAddress) property

HTTPPort

URL

LocalFileName

ParseIncomingData

When the connection is successful, the HTTPServerConnection event fires.

2. After the HTTPServerConnection event fires, call the GET method to retrieve the Web page identified by the URL.

After the data is retrieved and parsed, the FileClosed event fires, and data from these properties can now be read;

The file identified by the LocalFileName property

The HTMLPageTextWithTags property

The HTMLPageTextWithoutTags property

The ListBox objects named by AnchorListBoxName,

ImageFileListBoxName,

and TagListBoxName properties

(if anchors, images, and tags exist in the Web page)

The PacketReceived event

The FileClosed event

The code looks like this;

Form_Load

```
CIHTTP1.HostName = www.myhost.com
```

```
CIHTTP1.HTTPPort = 80
```

```
CIHTTP1.URL = http://www.progress.com/crescent/?HTTP/1.0
```

```
CIHTTP1.LocalFileName = C:\THEPAGE.HTML
```

```
CIHTTP1.ParseIncomingData = True
```

```
CIHTTP1.ConnectToHTTPServer
```

```
.
```

```
HTTPServerConnection_Event
```

```
CIHTTP1.GET
```



Posting Data to a Web Site

Related Topics

This topic illustrates how you can use CIHTTP to post data to a Web site.

1. Set these properties, then call the ConnectToHTTPServer;
HostName (or HostAddress) property
HTTPPort
URL
LocalFileName

When the connection is successful, the HTTPServerConnection event fires.

2. Call the POST method to post the data to the Web site specified by the URL property.

Form_Load

```
CIHTTP1.HostName = www.myhost.com  
CIHTTP1.HTTPPort = 80  
CIHTTP1.URL = http://www.progress.com/crescent/?HTTP/1.0  
CIHTTP1.LocalFileName = C:\THEPAGE.HTML  
CIHTTP1.ConnectToHTTPServer
```

HTTPServerConnection_Event

```
CIHTTP1.POST
```

When the data is sent, the PacketSent event fires. The data or object that you post is appended to the Web location specified in the URL.

NOTE Many Web servers require that you have write privileges to perform a POST.

Getting Started With the CIHTTPControl

What is HTTP?

What Is the CIHTTP Control?

Programming Tasks

Retrieve a Web Page

Posting Data to a Web Site

What is HTTP?

What Is the CIHTTP Control?

Programming Tasks

Retrieve a Web Page

Posting Data to a Web Site

The Crescent InterNet HTTP Control Reference

[Getting Started With the CIHTTPControl](#)

[What Is the CIHTTP Control?](#)

[Programming Tasks](#)

[Retrieve a Web Page](#)

[Posting Data to a Web Site](#)

[The Crescent InterNet HTTP Control Reference](#)

[Getting Started With the CIHTTPControl](#)

[What is HTTP?](#)

[Programming Tasks](#)

[Retrieve a Web Page](#)

[Posting Data to a Web Site](#)

[The Crescent InterNet HTTP Control Reference](#)

[Getting Started With the CIHTTPControl](#)

[What is HTTP?](#)

[What Is the CIHTTP Control?](#)

[Retrieve a Web Page](#)

[Posting Data to a Web Site](#)

[The Crescent InterNet HTTP Control Reference](#)

[Getting Started With the CIHTTPControl](#)

[What is HTTP?](#)

[What Is the CIHTTP Control?](#)

[Programming Tasks](#)

[Posting Data to a Web Site](#)

[The Crescent InterNet HTTP Control Reference](#)

[Getting Started With the CIHTTPControl](#)

[What is HTTP?](#)

[What Is the CIHTTP Control?](#)

[Programming Tasks](#)

[Retrieve a Web Page](#)

[The Crescent InterNet HTTP Control Reference](#)



The Crescent Internet ToolPak HTTP Control

Related Topics

Control File

CIHTTP.OCX

Object Type

CIHTTP

Purpose

The Crescent Internet HTTP control (CIHTTP) enables you to write customized World Wide Web (WWW) client applications. You can use CIHTTP to retrieve text or binary data from a Web site, to retrieve data about a page at the Web site (such as date last updated), or to post text to a Web site from within an OLE container. CIHTTP also has the ability to parse the retrieved data. The control is intended to enhance not replace a Web browsers functionality.

Some typical applications that you might build with this control include:

Web Data Retrieval Applications

Use the CIHTTP control to access data from a Web site and place it in a spreadsheet or other customized database. For example, you might create an application that pulls stock data from the Internet and places it in an Excel spreadsheet.

Web Data Entry Applications

Create applications in which users enter data to a form, then use the CIHTTP control to post the data form to a Web site.

Web Crawler Applications

Create a Web Crawler that searches the Internet for Web sites that contain information relevant to specific keywords.

Properties

About	<u>AnchorListBoxName</u> c	Container
<u>EventState</u> c	Height	<u>HostAddress</u> c
<u>HostName</u> c	<u>HTMLPageTextWithoutTags</u> c	<u>HTMLPageTextWithTags</u> c
<u>HTTPPort</u> c	<u>ImageFileListBoxName</u> c	Index
Left	<u>LocalFileName</u> c	<u>MethodState</u> c
Name	Object	Parent
<u>ParseIncomingData</u> c	<u>ProxyServerAddress</u> c	<u>ProxyServerName</u> c
Tag	<u>TagListBoxName</u> c	Top
<u>URL</u> c	Visible	WhatsThisHelpID
Width	<u>WWWSiteName</u> c	ZOrder

Events

DragDrop	DragOver	<u>EventStateChanged</u> c
<u>FileClosed</u> c	<u>HTTPServerConnectionClosed</u> c	<u>HTTPServerConnection</u> c
<u>ListBoxesPopulated</u> c	<u>MethodStateChanged</u> c	<u>PacketReceived</u> c
<u>PacketSent</u> c	<u>SocketClosed</u> c	<u>TotalFileBytesReceived</u> c

WSAError c

Methods

CleanupConnection c

DragIcon

HEAD c

SendHTTPCommand c

ConnectToHTTPServer c

DragMode

Move

ShowWhatsThis

Drag

GET c

POST c

c A custom or modified property, method, or event.

AnchorListBoxName Property

Applies To

CIHTTP

Purpose

The AnchorListBoxName property sets the name of a listbox object that is bound to the CIHTTP control. When the control retrieves an HTML page from a Web site, the listbox is populated with a list of the anchors from the parsed HTML page.

Syntax

```
Set [Form.]CIHTTP.AnchorListBoxName = ObjectName
```

Data Type

Object

Usage

Read only at runtime

Comments

An anchor is a hypertext link. The link can be to another location within the document, to another document on the same Web server, or to a document on another Web server. The AnchorListBox object gathers the anchors in a given Web page into one location so that you can then programmatically use that information to jump to those sites.

NOTE Links to locations within the same document and links to other documents on the same server might not contain complete URL information (it might be a relative not absolute URL). When accessing these types of links, you might have to programmatically maintain/insert the URL.

If the data retrieved is a binary file (.exe or image file), CIHTTP ignores this property.

When the AnchorListBox, ImageListBox, and TagListBox are populated, the ListBoxesPopulated event fires.

See Also

ImageFileListBoxName, ListBoxesPopulated, TagListBoxName

EventState Property

Applies To

CIHTTP

Purpose

The EventState Property returns the event currently in progress.

Syntax

[Integer%] = [Form.] CIHTTP.EventState

Data Type

Integer

Usage

Read/Write at runtime

Comments

CIHTTP updates EventState when an event fires. When the EventState Property changes, CIHTTP fires the EventStateChanged event. The valid EventState Property values are listed in the following table and are also in the CITPAK.BAS file.

This event. . .	has this constant. . .	with this integer value. . .
ListBoxesPopulated	CIHTTP_LBPOP	120
HTTPServerConnection	CIHTTP_CONN	121
SocketClosed	CIHTTP_SCLOSED	122
HTTPServerConnectionClosed	CIHTTP_CONNCLOSED	123
FileClosed	CIHTTP_FCLOSED	124

You can test the EventState to initiate an action or control program flow.

See Also

EventStateChanged

HostAddress Property

Applies To

CIHTTP

Purpose

The HostAddress property sets or returns the IP address of the Web server.

Syntax

```
[Form.] CIHTTP.HostAddress [ = String$]
```

Data Type

String

Usage

Read/Write at design time

Comments

The HostAddress identifies the host to the network (the Internet). HostAddress consists of the network number and the local host. An example of an IP address is:

```
198.137.64.1
```

To connect to a Web server you must provide either the HostName or the HostAddress property. If you provide both, CIHTTP uses the HostName. In addition, if you are connected to the Internet via a Proxy server (known as a firewall system), you must also provide the ProxyServerAddress or ProxyServerName. If you provide both, CIHTTP uses the ProxyServerName.

See Also

[ConnectToHTTPServer](#), [HostName](#), [ProxyServerAddress](#), [ProxyServerName](#)

HostName Property

Applies To

CIHTTP

Purpose

The HostName property sets the name of the Web server.

Syntax

```
[Form.] CIHTTP.HostName [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime

Comments

The HostName is the name that identifies an HTTP server to the Internet. Depending on your connection to the Internet, the HostName might contain simply a machine name like `crescentserver`, or it can be a fully qualified Internet name that follows this format: `machine.organizationname.com`. Contact your site administrator to determine the correct format to supply.

Set HostAddress or HostName before calling the ConnectToHTTPServer method. If you provide both, CIHTTP uses the HostName. In addition, if you are connected to the Internet via a Proxy server (known as a firewall system), you must also provide the ProxyServerAddress or ProxyServerName. If you provide both, CIHTTP uses the ProxyServerName.

See Also

[ConnectToHTTPServer](#), [HostAddress](#), [ProxyServerAddress](#), [ProxyServerName](#)

HTMLPageTextWithoutTags Property

Applies To

CIHTTP

Purpose

The HTMLPageTextWithoutTags property returns the parsed Web page without HTML formatting tags.

Syntax

```
String$ = [Form.] CIHTTP.HTMLPageTextWithoutTags
```

Data Type

String

Usage

Read only at runtime

Comments

Use the contents of the HTMLPageWithTags property to display the HTML page excluding the HTML formatting tags.

See Also

HTMLPageTextWithTags

HTMLPageTextWithTags Property

Applies To

CIHTTP

Purpose

The HTMLPageTextWithTags property returns the parsed Web page with HTML formatting tags.

Syntax

```
String$ = [Form.]CIHTTP.HTMLPageTextWithTags
```

Data Type

String

Usage

Read only at runtime

Comments

Use the contents of the HTMLPageWithTags property to display the HTML page including the tags. You must also set the ParseIncomingData property to true.

See Also

HTMLPageTextWithoutTags, ParseIncomingData

HTTPPort Property

Applies To

CIHTTP

Purpose

The HTTPPort property sets the number of the port used for HTTP communications.

Syntax

```
[Form.] CIHTTP.HTTPPort [ = Integer% ]
```

Data Type

Integer

Usage

Read/Write at design time and runtime

Comments

The default is 80.

ImageFileListBoxName Property

Applies To

CIHTTP

Purpose

The ImageFileListBoxName property contains the name of a listbox control that is bound to the CIHTTP control. When the control retrieves an HTML page from a Web site, the listbox is populated with the list of image files from the parsed HTML page.

Syntax

```
Set [Form.]CIHTTP.ImageFileListBoxName = ObjectName
```

Data Type

Object

Usage

Read only at runtime

Comments

CIHTTP ignores this property if the data retrieved is not an HTML page.

When the AnchorListBox, ImageListBox, and TagListBox are populated, the ListBoxesPopulated event fires.

You must also set the ParseIncomingData property to true.

See Also

AnchorListBoxName, ListBoxesPopulated, ParseIncomingData, TagListBoxName

LocalFileName Property

Applies To

CIHTTP

Purpose

The LocalFileName property sets a path and filename on the local machine.

Syntax

```
[Form.] CIHTTP.LocalFileName [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime

Comments

Use the local file to store data retrieved from the Web via the GET method.

MethodState Property

Applies To

CIHTTP

Purpose

The MethodState Property returns the method currently in progress.

Syntax

```
[Integer%] = [Form.] CIHTTP.MethodState
```

Data Type

Integer

Usage

Read/Write at runtime

Comments

CIHTTP updates MethodState when you call a method like GET, HEAD, or POST. When the MethodState Property changes, CIHTTP fires the MethodStateChanged event. The valid MethodState Property values are listed in the following table and are also in the CITPAK.BAS file.

This method...	has this constant...	with this integer value...
GET	CIHTTP_GET	14
HEAD	CIHTTP_HEAD	15
POST	CIHTTP_POST	16

You can test the MethodState to initiate an action or control program flow.

See Also

MethodStateChanged

ParseIncomingData Property

Applies To

CIHTTP

Purpose

The ParseIncomingData property sets whether or not to parse the data retrieved from a Web site.

Syntax

```
[Form.] CIHTTP.ParseIncomingData [ = Boolean% ]
```

Data Type

Boolean Integer

Usage

Read/Write at design time and runtime

Comments

If the data is a file, this property is ignored.

Value

True

False

Behavior

CIHTTP parses the data based on the HTML tags.

CIHTTP does not parse the data; it assumes the data is binary.

ProxyServerAddress Property

Applies To

CIHTTP

Purpose

The ProxyServerAddress sets the IP address of the *proxy server* (the server that acts as the firewall).

Syntax

```
[Form.] CIHTTP.ProxyServerAddress [ = String$ ]
```

Data Type

String

Usage

Read/Write at design time

Comments

The ProxyServerAddress identifies the proxy server to the Internet. The ProxyServerAddress, like the HostAddress, consists of the network number and the local host number. When your site uses a firewall system, you must provide either the ProxyServerAddress or the ProxyServerName as well as the HostName or HostAddress to retrieve objects/documents from an Web server.

An example of an IP address is:

198.137.64.1

NOTE If you provide both a `HostName` and `HostAddress`, `CIHTTP` uses the `HostName`. If you provide both a `ProxyServerName` and a `ProxyServerAddress`, `CIHTTP` uses the `ProxyServerName`.

See Also

[ConnectToHTTPServer](#), [HostAddress](#), [HostName](#), [ProxyServerName](#)

ProxyServerName Property

Applies To

CIHTTP

Purpose

The ProxyServerName property sets the name of the *proxy server* (the server that acts as the firewall).

Syntax

```
[Form.] CIHTTP.ProxyServerName [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime

Comments

The ProxyServerAddress identifies the proxy server to the Internet. When you are connected to the Internet via a Proxy server (known as a firewall system), you must provide the ProxyServerAddress or ProxyServerName. As with HostName and HostAddress, if you provide both, CIHTTP uses the ProxyServerName.

See Also

[ConnectToHTTPServer](#), [HostAddress](#), [ProxyServerAddress](#)

TagListBoxName Property

Applies To

CIHTTP

Purpose

The TagListBoxName property sets the name of a listbox control that is bound to the CIHTTP control. When the control retrieves an HTML page from a Web site, the listbox is populated with a list of the tags from the parsed HTML page.

Syntax

```
Set [Form.]CIHTTP.TagListBoxName [ = ObjectName]
```

Data Type

Object

Usage

Read only at runtime

Comments

A tag is a code for a format to be applied to the Web page, for example <title>. The TagListBox object gathers the tags in a given Web page into one location so that you can view the styles in the document. This information can be useful to search for uses of specific tags.

When the AnchorListBox, ImageListBox, and TagListBox are populated, the ListBoxesPopulated event fires.

If the data retrieved is a binary file (.exe or image file), CIHTTP ignores this property.

You must also set the ParseIncomingData property to true.

See Also

[AnchorListBoxName](#), [ImageFileListBoxName](#), [ListBoxesPopulated](#), [ParseIncomingData](#)

URL Property

Applies To

CIHTTP

Purpose

The URL property sets the Uniform Resource Locator of the current request.

Syntax

```
[Form.]CIHTTP.URL[ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime

Comments

A Uniform Resource Locator identifies the location of an object on the Internet using a standard address syntax. It contains information in the address that indicates how you might access the object. The following example illustrates the URL for the Progress Software Web page:

URL `http://www.progress.com`

WWWSiteName Property

Applies To

CIHTTP

Purpose

The WWWSiteName property contains the name of the current Web page. It is extracted from the <title> tag of the retrieved HTML page if one exists.

Syntax

```
String$ = [Form.]CIHTTP.WWWSiteName
```

Data Type

String

Usage

Read only at runtime

See Also

TagListBoxName

EventStateChanged Event

Applies To

CIHTTP

Purpose

The EventStateChanged event fires when the value of the EventState Property changes.

Syntax

```
Sub CIHTTP_EventStateChanged(ByVal state As Integer)
```

Comments

The EventState Property records the state of CIHTTP; its value changes in response to other CIHTTP events. You can test the EventState Property to determine a state change and thus initiate an action, or control program flow. The valid EventState Property values are listed in the following table and are also in the CITPAK.BAS file.

This event. . .	has this constant. . .	with this integer value. . .
ListBoxesPopulated	CIHTTP_LBPOP	120
HTTPServerConnection	CIHTTP_CONN	121
SocketClosed	CIHTTP_SCLOSED	122
HTTPServerConnectionClosed	CIHTTP_CONNCLOSED	123
FileClosed	CIHTTP_FCLOSED	124

See Also

EventState

FileClosed Event

Applies To

CIHTTP

Purpose

The FileClosed event fires when the file stream (specified by the LocalFileName property) on the client closes.

Syntax

```
Sub CIHTTP_FileClosed()
```

Comments

The FileClosed event indicates that communications with the Web server has terminated, and that all data has been retrieved and parsed.

See Also

LocalFileName

HTTPServerConnection Event

Applies To

CIHTTP

Purpose

The HTTPServerConnection event fires when CIHTTP establishes a connection to a Web server.

Syntax

```
Sub CIHTTP_HTTPServerConnection()
```

Comments

Once the HTTPServerConnection event fires, your application can start to interact with the Web server by invoking the GET, POST, or the SendHTTPCommand methods.

Because the HTTPServerConnection event fires when a connection is established, the HTTPServerConnection_Event module is a good module to place any initial code.

See Also

GET, HTTPServerConnectionClosed, POST, SendHTTPCommand

HTTPServerConnectionClosed Event

Applies To

CIHTTP

Purpose

The HTTPServerConnectionClosed event fires when a connection to a Web server terminates.

Syntax

```
Sub CIHTTP_HTTPServerConnectionClosed()
```

Comments

Connections are terminated at the request of the Web server.

See Also

HTTPServerConnection

ListBoxesPopulated Event

Applies To

CIHTTP

Purpose

The ListBoxesPopulated event fires when the AnchorListBox, ImageListBox, and TagListBox objects have been filled.

Syntax

```
Sub CIHTTP_ListBoxesPopulated()
```

Comments

The event fires once after all of the ListBoxes are populated.

See Also

AnchorListBoxName, ImageFileListBoxName, ListBoxesPopulated, TagListBoxName

MethodStateChanged Event

Applies To

CIHTTP

Purpose

The MethodStateChanged event fires when the value of the MethodState Property changes.

Syntax

```
Sub CIHTTP_MethodStateChanged(ByVal state As Integer)
```

Comments

The MethodState Property records the state of CIHTTP; its value changes in response to other CIHTTP events. You can test the MethodState Property to determine a state change and thus initiate an action, or control program flow. The valid MethodState Property values are listed in the following table and are also in the CITPAK.BAS file.

This method...	has this constant...	with this integer value...
GET	CIHTTP_GET	14
HEAD	CIHTTP_HEAD	15
POST	CIHTTP_POST	16

See Also

MethodState

PacketReceived Event

Applies To

CIHTTP

Purpose

The PacketReceived event fires when CIHTTP receives a packet from the connected Web server.

Syntax

```
Sub CIHTTP_PacketReceived(ByVal Packet As String, ByVal bytes_in As Integer)
```

Comments

The *bytes_in* value is the number of bytes in the current packet.

The *Packet* contains data received from the server; it can contain a Web page or header information from a Web page depending on the method used to obtain the packet. The *bytes_in* value is the number of bytes in the current packet.

See Also

Event PacketSent, GET, HEAD, SendHTTPCommand, TotalFileBytesReceived

PacketSent Event

Applies To

CIHTTP

Purpose

The PacketSent event fires when CIHTTP sends a packet to a Web server.

Syntax

```
Sub CIHTTP_PacketSent(ByVal bytes_out As Integer)
```

Comments

The value of *bytes_out* is the number of bytes in the current packet. Packets are sent in response to the all of the methods except: CleanupConnection.

See Also

GET, HEAD, PacketReceived, POST, SendHTTPCommand

SocketClosed Event

Applies To

CIHTTP

Purpose

The SocketClosed event fires when the socket closes.

Syntax

```
Sub CIHTTP_SocketClosed()
```

Comments

The socket should remain open during program execution. If the SocketClosed event fires while your application is running, an error condition has occurred.

See Also

CleanupConnection

TotalFileBytesReceived Event

Applies To

CIHTTP

Purpose

The TotalFileBytesReceived event fires repeatedly as packets are received from the server.

Syntax

```
Sub CIHTTP_TotalFileBytesReceived(ByVal bytes_in As Long)
```

Comments

The *bytes_in* value is the total number of bytes received during the file transfer. It increments each time a packet is received. Use *bytes_in* as the byte counter for the file transfer.

See Also

PacketReceived

WSAError Event

Applies To

CIHTTP

Purpose

The WSAError event fires when CIHTTP receives a Winsock error.

Syntax

```
Sub CIHTTP_WSAError(ByVal error_number As Integer)
```

Comments

Use the WSAError event to monitor the Winsock activity. The *error_number* identifies the Winsock error that occurred. These errors are listed in the CITPAK.BAS file.

See Also

CITPAK.BAS

CleanupConnection Method

Applies To

CIHTTP

Purpose

The CleanupConnection method closes the socket and cleans up Winsock.

Syntax

```
CIHTTP.CleanupConnection
```

Data Type

None

Comments

By default CIHTTP opens the socket when you invoke the ConnectToHTTPServer method and closes it when the Web server notifies the client that the transaction is finished. A transaction can be finished when the servers got your data from the POST method, or its sent all of the data in response to the GET or HEAD methods.

Use CleanupConnection to clear errors that might occur when a method fails, or in other cases where the socket does not close properly and you need to restart the connection process. CleanupConnection fires the SocketClosed event.

NOTE Use this method only when all else fails.

See Also

ConnectToHTTPServer, SocketClosed, WSAError

ConnectToHTTPServer Method

Applies To

CIHTTP

Purpose

The ConnectToHTTPServer method establishes a connection to a Web server.

Syntax

```
nResult = CIHTTP.ConnectToHTTPServer
```

Data Type

Integer

Comments

You must establish a connection to the Web server before CIHTTP can perform any send or retrieve functions. To establish a connection, you must provide the `HostName` **or** `HostAddress` property and the `HTTPPort` properties. In addition, when your system is connected to the Internet via a firewall system, you must also provide the `ProxyServerAddress` **or** the `ProxyServerName` before you can connect.

When the connection attempt succeeds, *nResult* is an integer that represents the socket number opened by the ConnectToHTTPServer method. The return of a socket number does not indicate successful connection; the `HTTPServerConnection` event fires when the connection attempts succeeds. When the connection attempt fails, *nResult* is 0 and the `WSAError` event fires.

See Also

[HostAddress](#), [HostName](#), [HTTPPort](#), [HTTPServerConnection](#), [ProxyServerAddress](#), [ProxyServerName](#), [WSAError](#)

GET Method

Applies To

CIHTTP

Purpose

The GET method retrieves the Web page from the location specified by the URL property.

Syntax

CIHTTP.GET

Data Type

None

Comments

Before invoking GET, you must be connected to a Web server (via the ConnectToHTTPServer method), and you must set the URL property.

GET updates the LocalFile, the HTMLPageTextWithTags, the AnchorListBox, the ImageFileListBox, the TagsListBox objects and the PacketReceived event.

See Also

[AnchorListBoxName](#), [HTMLPageTextWithTags](#), [ImageFileListBoxName](#), [LocalFileName](#), [PacketReceived](#), [TagListBoxName](#), [URL](#)

HEAD Method

Applies To

CIHTTP

Purpose

The HEAD method retrieves page information from the location specified by the URL property.

Syntax

CIHTTP.HEAD

Data Type

None

Comments

Set the URL property and before calling the HEAD method. HEAD updates the LocalFile object and the PacketReceived event.

See Also

LocalFileName, PacketReceived

POST Method

Applies To

CIHTTP

Purpose

The POST method lets you send text to a Web server.

Syntax

CIHTTP.POST

Data Type

None

Comments

Set the URL property before calling the POST method. The text that you post is appended to the Web location specified by the URL property. Many Web servers require that you have write privileges to perform a POST.

The POST method can be used to create data entry applications. For example, you can create applications that let customers request catalogs or place orders. When the customer fills out the data entry form, the POST method sends the completed form back to the Web server.

SendHTTPCommand Method

Applies To

CIHTTP

Purpose

The SendHTTPCommand method lets you issue HTTP commands other than GET, HEAD, and POST.

Syntax

```
CIHTTP.SendHTTPCommand(Packet)
```

Data Type

String

Comments

Before invoking SendHTTPCommand, set the HostAddress **or** the HostName property, and the URL. In addition, if your system is connected to the Internet via a firewall system, you must also set the ProxyServerAddress **or** the ProxyServerName before invoking the SendHTTPCommand method.

The *Packet* is the HTTP command that you want to issue.

Note also that when the SendHTTPCommand sends the packet, the PacketSent event fires.

See Also

GET, HEAD, HostAddress, HostName, PacketSent, POST, ProxyServerAddress, ProxyServerName, URL



General Tab

Applies To

CIHTTP

This custom tab lets you set these custom properties:

HostAddress

HostName

HTTPPort

LocalFileName

ParseIncomingData

URL



Sound Events Tab

Applies To

CIHTTP

This custom tab lets you assign sounds (via .WAV files) to custom events. The following lists the hidden properties that define the sound object and the event to which they are applied:

Event

Hidden Property

FileClosed

FileClosedWAV

HTTPServerConnection

HTTPServerConnectionWAV

HTTPServerConnectionClosed

HTTPServerConnectionClosedWAV

ListBoxesPopulated

ListBoxesPopulatedWAV

PacketReceived

PacketReceivedWAV

PacketSent

PacketSentWAV

SocketClosed

SocketClosedWAV

WSAError

WSAErrorWAV

You can set the sound associations at design time or runtime. You can test the sounds from within the property page. At runtime, an event is fired and a sound is played asynchronously where an association is made.

To associate a sound, you can type in the full qualified path for any events you want to associate or click the Browse... button and select it from the dialog box. After you enter a sound, you can click the test button to hear it. If the sound file is invalid or missing you will hear nothing when you click Test.

