VB5CGI Object Reference

Reference

- **Environment Variable**
- **HTTP Header**
- **Cookie**
- **Hit Counter**
- **String Conversion**
- **Information & Communication**
- **Debugging**

VB5CGI Example

VB5HTML Object
VB5DB Object
VB5-CGI Objects Overview

VB5CGI Object Reference

Reference

VB5CGI Object Reference

Reference

VB5CGI Object Reference

Reference

VB5CGI Object Reference

Reference

VB5CGI Object Reference

Reference

VB5CGI Object Reference

Reference

VB5CGI Object Reference

Reference

EnvAuthType

Returns the Script environment variable AUTH_TYPE.

Syntax

Property EnvAuthType As String (read-only)

Example

```
Dim str As String
str = CGI.EnvAuthType
```

EnvContentLength

Returns the Script environment variable CONTENT_LENGTH as type long.

Syntax

Property EnvContentLength As Long (read-only)

Example

```
Dim length As Long
length = CGI.EnvContentLength
```

EnvContentType

Returns the Script environment variable CONTENT_TYPE.

Syntax

Property EnvContentType As String (read-only)

Example

```
Dim str As String
str = CGI.EnvContentType
```

EnvGatewayInterface

Returns the Server environment variable GATEWAY_INTERFACE.

Syntax

Property EnvGatewayInterface As String (read-only)

Example

```
Dim str As String
str = CGI.EnvGatewayInterface
```

EnvHTTPAccept

Returns the Client environment variable HTTP_ACCEPT.

Syntax

Property EnvHTTPAccept As String (read-only)

Example

```
Dim str As String
str = CGI.EnvHTTPAccept
```

EnvHTTPReferer

Returns the Client environment variable HTTP_REFERER.

Syntax

Property EnvHTTPReferer As String (read-only)

Example

```
Dim str As String
str = CGI.EnvHTTPReferer
```

EnvHTTPUserAgent

Returns the Client environment variable HTTP_USER_AGENT.

Syntax

Property EnvHTTPUserAgent As String (read-only)

Example

```
Dim str As String
str = CGI.EnvHTTPUserAgent
```

EnvPathInfo

Returns the Script environment variable PATH_INFO.

Syntax

Property EnvPathInfo As String (read-only)

Example

```
Dim str As String
str = CGI.EnvPathInfo
```

EnvPathTranslated

Returns the Script environment variable PATH_TRANSLATED.

Syntax

Property EnvPathTranslated As String (read-only)

Example

```
Dim str As String
str = CGI.EnvPathTranslated
```

EnvQueryString

Returns the Script environment variable QUERY_STRING.

Syntax

Property EnvQueryString As String (read-only)

Example

```
Dim str As String
str = CGI.EnvQueryString
```

EnvRemoteAddr

Returns the Client environment variable REMOTE_ADDR.

Syntax

Property EnvRemoteAddr As String (read-only)

Example

```
Dim str As String
str = CGI.EnvRemoteAddr
```

EnvRemoteHost

Returns the Client environment variable REMOTE_HOST.

Syntax

Property EnvRemoteHost As String (read-only)

Example

```
Dim str As String
str = CGI.EnvRemoteHost
```

EnvRemoteIdent

Returns the Client environment variable REMOTE_IDENT.

Syntax

Property EnvRemoteIdent As String (read-only)

Example

```
Dim str As String
str = CGI.RemoteIdent
```

EnvRemoteUser

Returns the Client environment variable REMOTE_USER.

Syntax

Property EnvRemoteUser As String (read-only)

Example

```
Dim str As String
str = CGI.EnvRemoteUser
```

EnvRequestMethod

Returns the Script environment variable REQUEST_METHOD (usually GET or POST).

Syntax

Property EnvRequestMethod As String (read-only)

Example

```
Dim str As String
str = CGI.EnvRequestMethod
```

EnvScriptName

Returns the Script environment variable SCRIPT_NAME.

Syntax

Property EnvScriptName As String (read-only)

Example

```
Dim str As String
str = CGI.EnvScriptName
```

EnvServerName

Returns the Server environment variable SERVER_NAME.

Syntax

Property EnvServerName As String (read-only)

Example

```
Dim str As String
str = CGI.EnvServerName
```

EnvServerPort

Returns the Server environment variable SERVER_PORT.

Syntax

Property EnvServerPort As Long (read-only)

Example

```
Dim port As Long
port = CGI.EnvServerPort
```

EnvServerProtocol

Returns the Server environment variable SERVER_PROTOCOL.

Syntax

Property EnvServerProtocol As String (read-only)

Example

```
Dim str As String
str = CGI.EnvServerProtocol
```

EnvServerSoftware

Returns the Server environment variable SERVER_SOFTWARE.

Syntax

Property EnvServerSoftware As String (read-only)

Example

```
Dim str As String
str = CGI.EnvServerSoftware
```

GetEnvValue

Returns an environment variable string value.

Syntax

Function GetEnvValue(ByVal EnvironVarName As String) As String

Parameters

EnvironVarName : Environment variable string.

Return Value

Environment variable value string.

Example

```
Dim str as String
str = CGI.GetEnvVarValue("HTTP_COOKIE")
```

If the HTTP_COOKIE environment variable exists, str contains the value. Otherwise str contains an empty string.

ServerStatus

Returns/Sets the HTTP-header server status (default="Status: 200 OK").

Syntax

Property ServerStatus As String

Example

```
CGI.ServerStatus = "HTTP/1.0 200 OK"
```

This example replaces the server status from default "Status: 200 OK" to "HTTP/1.0 200 OK" which also works with most available Web servers.
If you expierence problems, try to change the *ServerStatus* value to an empty string:

```
CGI.ServerStatus = ""
```

ContentType

Returns/Sets the HTTP-header content type (default="text/html").

Syntax

Property ContentType As String

Example

```
CGI.ContentType = "text/plain"
```

This example will replace the HTTP-header entry (also called MIME-header) from default "Content-Type: text/html" to "Content-Type: text/plain".

GetHTTPHeader

Returns an HTTP-header string (includes: Content-Type; optionally: Pragma, Cookie data and HTTPHeaderText).

Syntax

Function GetHTTPHeader() As String

Return Value

HTTP-header string.

Example

```
Dim str As String
str = CGI.GetHTTPHeader()
```

The variable str contains by default the following string:

```
CONTENT-TYPE: text/html
PRAGMA: NO-CACHE
[cr&lf]
```

HTTPHeaderText

Returns/Sets additional HTTP-header text.

Syntax

Property HTTPHeaderText As String

Example

```
CGI.HTTPHeaderText = "Status: 400 Bad Request"
```

This example adds the error message "Status: 400 Bad Request" to the HTTP-header.

PragmaNoCache

Returns/Sets the HTTP-header "Pragma: no-cache" (default=True=not cached).

Syntax

Property PragmaNoCache As Boolean

Example

```
CGI.PragmaNoCache = False
```

This example inserts the text "Pragma: no-cache" to the HTTP-header, which sets the page to 'cached' (by default the page gets not cached).

GetCookieValue

Returns an HTTP client cookie value.

Syntax

Function GetCookieValue(ByVal CookieName As String) As String

Parameters

CookieName : Cookie name.

Return Value

The cookie value.

Example

```
Dim str As String
CGI.SetCookie "MyCookieName", "MyCookieValue"
str = CGI.GetCookieValue("MyCookieName")
```

The variable str contains the value "MyCookieValue" (if the browser allows cookies).

SetCookie

Sets an HTTP client cookie (multiple cookies can be set).

Syntax

Sub SetCookie(ByVal Name As String, ByVal Value As String, [ByVal Expires As String], [ByVal Domain As String], [ByVal Path As String = "/"], [ByVal Secure As Boolean])

Parameters

Name : Cookie name.
Value : Cookie value.
Expires : Expiring date in the format "Wdy, DD-Mon-YY HH:MM:SS GMT", default=the cookie will expire when the user's session ends.
Domain : Domain name, e.g. "www.myserver.com", default=""=actual server.
Path : Path name, e.g. "/mydocs/new", default="/"=all directories.
Secure : Transmit in secure mode (only possible if server supports SSL), default=False.

Example

```
Dim str As String
CGI.SetCookie "MyCookieName", "MyCookieValue"
str = CGI.GetCookieValue("MyCookieName")
```

The variable str contains the value "MyCookieValue" (if the browser allows cookies).

GetHitCounterInc

Returns and increments [+1] the hit counter value for a given counter name [DefaultCounter].

Syntax

Function GetHitCounterInc([ByVal CounterName As String = "DefaultCounter"], [ByVal IncrementValue As Long = 1], [ByVal UseRegistry As Boolean]) As Long

Parameters

CounterName : Name of the counter name (max 255 characters), default="DefaultCounter".
IncrementValue : The value, by which the counter will be incremented (0..2,147,483,647), default = 1.
UseRegistry : If False (default), the hit counter name and value will be first time created and saved in the file "HitCount.ini", which is located in the same directory as the VB5CGI.DLL.
If True, the hit counter name and value will be first time created and saved in the Windows Registry under "HKEY_CURRENT_USER\Software\VB and VBA Program Settings\VB5CGI\HitCounter\ **CounterName=n**.
Note: Under Windows NT4, using the Windows Registry as hit counter repository, will not work correctly with all Server software. In this case, use the file, which is the default setting.

Return Value

New hit counter value. If used the first time, it will return 1.

Example

```
Dim count As Long
count = CGI.GetHitCounterInc("MyCounter")
```

The variable count contains the last counter value + 1.
In this example, the new counter value will be saved in the file "HitCount.ini", which is located in the same directory as the VB5CGI.DLL.

SetHitCounterValue

Sets the hit counter value [0] for a given counter [DefaultCounter].

Syntax

Sub SetHitCounterValue([ByVal CounterName As String = "DefaultCounter"], [ByVal Value As Long = 0], [ByVal UseRegistry As Boolean])

Parameters

CounterName : Name of the counter name (max 255 characters), default="DefaultCounter".
Value : New counter value, default=0=reset hit counter.
UseRegistry : If False (default), the hit counter name and value will be first time created and saved in the file "HitCount.ini", which is located in the same directory as the VB5CGI.DLL.
If True, the hit counter name and value will be first time created and saved in the Windows Registry under "HKEY_CURRENT_USER\Software\VB and VBA Program Settings\VB5CGI\HitCounter\CounterName=n.
Note: Under Windows NT4, using the Windows Registry as hit counter repository, will not work correctly with all Server software. In this case, use the file, which is the default setting.

Example

```
CGI.SetHitCounterValue("MyCounter", 1000)
```

This example sets the hit counter "MyCounter" to the value of 1000.

GetPathFromURL

Converts URL-encoded path to DOS-type path ("/" to "\" and "+" to " ").

Syntax

Function GetPathFromURL(ByVal URL As String) As String

Parameters

URL : URL encoded (Unix-type) path string.

Return Value

DOS-type path string.

Example

```
Dim str As String
str = CGI.GetPathFromURL("/My+Dir/My+File")
```

The variable str contains the string "\My Dir\My File"

GetURLDecode

URL-decodes a string (%xx escapes to characters and "+" to " ").

Syntax

Function GetURLDecode(ByVal TextToDecode As String, [ByVal UseCustomDecodeEvent As Boolean]) As String

Parameters

TextToDecode : URL encoded string (containing escape characters).
UseCustomDecodeEvent : If True, instead of internal URL-decoding, an event will allow the developer to program his own decoding algorithm, default=False.

see: **CustomDecode Event**

Return Value

Plain text with replaced escape and '+' characters.

Example

```
Dim str As String
str = CGI.GetURLDecode("This+is+a+test%21")
```

The variable str contains the string "This is a test!".

CustomDecode Event

Occurs when using GetURLDecode or GetQueryString and parameter 'UseCustomDecodeEvent = true'.

Syntax

Event CustomDecode(TextToDecode As String)

Parameters

TextToDecode : URL-encoded text, which needs to be custom decoded.

Return Value

The event procedure has to return the decoded string in the 'TextToDecode' parameter.

Example

To be able to receive an event from an ActiveX-DLL, you have to create an additional class module. The following code shows the standard module, which is needed to create an .EXE file:

```
Private StartClass As New clsScript   'Create instance of the main script
class
Sub Main()
    StartClass.Start                  'Invoke the CGI script
End Sub
```

The startup procedure Main() of this standard module calls the Start method of the class module clsScript:

```
Private WithEvents CGI As VB5CGI.clsCGI         'Enable events for custom
decoding

'CGI start routine, which was invoked from the Sub Main
Public Sub Start()
    Set CGI = New VB5CGI.clsCGI                 'Create instance of the
VB5CGI object
    With CGI                                    'We use the VB5CGI object
        'Here comes your main script
    End With
End Sub

'This is the custom decode event procedure
Private Sub CGI_CustomDecode(TextToDecode As String)
    'Here comes your custom decode algorythm
End Sub
```

GetURLEncode

URL-encodes a string (characters not A-Z or a-z to %xx escapes and " " to "+").

Syntax

Function GetURLEncode(ByVal TextToEncode As String, [ByVal UseCustomEncodeEvent As Boolean]) As String

Parameters

TextToEncode : Plain text.
UseCustomEncodeEvent : If True, instead of internal URL-encoding, an event will allow the developer to program his own encoding algorithm, default=False.

see: **CustomEncode Event**

Return Value

URL encoded string

Example

```
Dim str As String
str = CGI.GetURLEncode("This is a test!")
```

The variable str contains the string "This+is+a+test%21".

CustomEncode Event

Occurs when using GetURLEncode and parameter 'UseCustomDecodeEvent = true'.

Syntax

Event CustomEncode(TextToEncode As String)

Parameters

TextToEncode : Plain text, which needs to be custom URL-encoded.

Return Value

The event procedure has to return the encoded string in the 'TextToEncode' parameter.

Example

To be able to receive an event from an ActiveX-DLL, you have to create an additional class module. The following code shows the standard module, which is needed to create an .EXE file:

```
Private StartClass As New clsScript   'Create instance of the main script
class
Sub Main()
    StartClass.Start                  'Invoke the CGI script
End Sub
```

The startup procedure Main() of this standard module calls the Start method of the class module clsScript:

```
Private WithEvents CGI As VB5CGI.clsCGI         'Enable events for custom
decoding

'CGI start routine, which was invoked from the Sub Main
Public Sub Start()
    Set CGI = New VB5CGI.clsCGI                 'Create instance of the
VB5CGI object
    With CGI                                    'We use the VB5CGI object
        'Here comes your main script
    End With
End Sub

'This is the custom encode event procedure
Private Sub CGI_CustomEncode(TextToEncode As String)
    'Here comes your custom encode algorithm
End Sub
```

GetURLFromPath

Converts DOS-type path to URL-encoded path ("\" to "/" and " " to "+").

Syntax

Function GetURLFromPath(ByVal Path As String) As String

Parameters

Path : DOS-type path string.

Return Value

URL encoded (Unix-type) path string.

Example

```
Dim str As String
str = CGI.GetURLFromPath("\MyDir\MyFile")
```

The variable str contains the string "/MyDir/MyFile".

SwapChar

Searches through string 'Source' replacing every instance of 'FromChar' with 'ToChar' (FromChar and ToChar may be any length).

Syntax

Sub SwapChar(Source As String, ByVal FromChar As String, ByVal ToChar As String)

Parameters

Source : String to modify.
FromChar : String which needs to be replaced.
ToChar : String to replace.

Return Value

The Source parameter (call by reference) contains the modified string.

Example

```
Dim str As String
str = "This is an example"
CGI.SwapChar(str, " ", "-")
```

The variable str contains the modified string "This-is-an-example". All spaces are replaced by dashes.

DumpEnvInfo

Dumps (submits to the browser) the environment variables and other information.

Syntax

Sub DumpEnvInfo()

Example

```
CGI.DumpEnvInfo
```

This line submits the environment variables and other information about the VB5CGI object.

GetDocRootPath

Returns the server HTML document root path (e.g. "C:\WebShare\WWWRoot").

Syntax

Function GetDocRootPath() As String

Return Value

Servers HTML document root path.

Example

```
Dim str As String
str = CGI.GetDocRootPath()
```

The variable str could contain "C:\WebShare\WWWRoot" (default path for the Personal Web Server).

GetPath

Returns the VB5CGI.DLL DOS-type path.

Syntax

Function GetPath([ByVal WithBackslash As Boolean = True]) As String

Parameters

WithBackslash : If True, returns the path with a trailing back slash ('\'), default=True.

Return Value

The absolute (including drive letter) DOS-type path of the VB5CGI object.

Example

```
Dim str As String
str = CGI.GetPath()
```

The variable str could contain "C:\Program Files\VB5-CGI Objects\".

GetQueryString

Returns the query string.

This is a low-level method to retrieve the input data, either from a GET or POST method. Unless you really need it, use the more convenient methods from the VB5HTML object.

Syntax

Public Function GetQueryString([ByVal AllowOnlyPOST As Boolean], [ByVal MaxQueryLen As Long = 2048], [ByVal URLDecoded As Boolean = True], [ByVal UseCustomDecodeEvent As Boolean])) As String

Parameters

AllowOnlyPOST : If True, only the POST method will be accepted, default=False (GET and POST are allowed).
Note: If AllowOnlyPOST=True and a GET method was invoked, the function result will be empty.
MaxQueryLen : Defines the maximum input length, of either the environment variable QUERY_STRING (GET method) or the standard input file (POST method), default=2048 characters.
Note: The length of the undecoded input can be longer than the decoded query string. If the query string is longer than MaxInputLen, only the left most part of the string will be returned.
URLDecoded : True (default), if the whole query string should be URL-decoded, otherwise False.
UseCustomDecodeEvent : If True, instead of internal URL-decoding, an event will allow the developer to program his own decoding algorithm, default=False.

see: **CustomDecode Event**

Return Value

POST method: Content of standard input file. By default URL-decoded.
GET method: Content of environment variable QUERY_STRING. By default URL-decoded.

Example

```
Dim str As String
str = CGI.GetQueryString(True, 1024)
```

The variable str contains the URL-encoded query string. In this example, the GetQueryString method accepts only data submitted by the POST method and returns only the first 1024 characters.

GetScriptName

Returns the filename of the running script [without path].

Syntax

Function GetScriptName([ByVal IncludePath As Boolean]) As String

Parameters

IncludePath : If True, returns the script name including the path, otherwise returns only the script name, default=False.

Return Value

Script name with/without the path (if from the server supported).

Example

```
Dim str As String
str = CGI.GetScriptName()
```

The variable str could contain the string "MyScript.EXE".

StdInput

Returns the standard input file contents (used for POST method).

This is a very low-level functionality. Unless you really need it, use the more convenient methods from the VB5HTML object.

See also: **GetQueryString**

Syntax

Function StdInput(ByVal BufferLength As Long) As String

Parameters

BufferLength : Number of expected characters.

Return Value

Standard input file string.

Example

```
Dim str As String
str = CGI.StdInput(CGI.EnvContentLength)
```

This example shows, how to retrieve a query string from a POST method.
The variable str contains the query string, if a POST method was invoked

StdOutput

Sends a string to the standard output file.

This is a very low-level functionality. Unless you really need it, use the more convenient methods from the VB5HTML object.

Syntax

Sub StdOutput(ByVal OutputText As String)

Parameters

OutputText : HTTP-header and HTML-text to submit.

Example

```
Dim msg As String

msg = "<HTML>" & _
        "<TITLE>Test</TITLE>" & _
        "<BODY>Hello World</BODY>" & _
      "</HTML>"

CGI.StdOutput CGI.GetHTTPHeader & msg
```

This example outputs a page (including HTTP-header with the function GetHTTPHeader) with the title 'Test' and the content 'Hello World'

DebugLogMode

Returns/Sets the CGI script debug file logging mode (default=cgiDebugOff).

Syntax

Property DebugLogMode As DebugEnum

The **DebugLogMode** property can have one or multiple DebugEnum constant values:
- cgiDebugAll: Write all possible debug information to the debug log file
- cgiDebugEnvVars: Write all environment variables to the debug log file
- cgiDebugExecTime: Write total script execution time to the debug log file
- cgiDebugInput: Write input data (from GET or POST method) to the debug log file
- cgiDebugOff: Disable debugging mode (default)
- cgiDebugOutput: Write output data to the debug log file

Example

```
CGI.DebugLogMode = cgiDebugInput + cgiDebugExecTime
```

This example sets the compiled script in debug mode and writes information about the input data (from a GET or POST method) and total execution time to the debug log file.
By default, without changing the DebugLogFile property, the log filename is "[VB5CGI.DLL path]\CGIdebug.log" and will be created on first time use.

DebugLogFile

Returns/Sets the debug log filename (default=[dll-path]\CGIdebug.log).

Syntax

Property DebugLogFile As String

Example

```
CGI.DebugLogFile = "C:\temp\mylogfile.txt"
```

This example changes the default log file path from "[VB5CGI.DLL]\CGIdebug.log" to "C:\temp\mylogfile.txt".

DebugLogAddEnvVar

Returns/Sets additional environment variable for the debug log file.

Syntax

Property DebugLogAddEnvVar As String

Example

```
CGI.DebugLogAddEnvVar = "DOCUMENT_URI"
```

If the debug mode is enabled (property DebugLogMode <> cgiDebugOff), this example will add the "DOCUMENT_URI" environment variable name and value to the debug log file, when running the compiled script.
This variable will be written to the log file, regardless of the cgiDebugEnvVars setting.

VB5CGI example

The following Visual Basic source code (demoCGI.bas) shows a simple CGI script, which uses the VB5CGI object. This example was
tested with the Microsoft Personal Web Server version 1.0a and 2.0 and the IIS3.
If the script gets called without any Query String (e.g. http://localhost/scripts/demoCGI.exe), a page with the servers environment variables and object information will be shown.
Otherwise, the script shows the URL decoded query string from the GET method, and if the browser allows cookies, the second cookie value "Value2" will be shown. A page counter informs about the number of times this script was run. The page title shows the script name including path.

```
Private CGI As New VB5CGI.clsCGI                'Instance the VB5CGI
Object (needs VB5CGI.DLL)

Sub Main()
Dim msg As String                               'String variable for HTML
text

    With CGI                                    'We use the VB5CGI object
        .DebugLogMode = cgiDebugAll             'Enable debugging to file
CGIdebug.log
        If Len(.GetQueryString) > 0 Then        'Did we receive a Query
String?
            .SetCookie "Name1", "Value1"        'Send the first cookie
            .SetCookie "Name2", "Value2"        'Send the second cookie
            msg = "<HTML><TITLE>VB5HTML demonstration ["
& .GetScriptName(True) & "]</TITLE>" & _
                  "<BODY>The Query String you sent with the "
& .EnvRequestMethod & _
                  " method is: " & .GetQueryString() & "<BR><P>" & _
                  "If you allow cookies, the second cookie value will be
next time: " & _
                  .GetCookieValue("Name2") & "<BR><P>" & _
                  "Page access: " & .GetHitCounterInc(App.EXEName) &
"</BODY></HTML>"
            .StdOutput .GetHTTPHeader & msg     'Submit the page with the
HTTP-header
        Else
            .DumpEnvInfo                        'Dump the environment
variables and VB5CGI information
        End If
    End With
End Sub
```

The following HTML text was used (demoCGI.htm) to invoke the script:

```
<HTML>
<HEAD>
  <TITLE>VB5-CGI Objects demo: VB5CGI object</TITLE>
</HEAD>

<BODY BGCOLOR="#FFFFFF">
<H2>VB5-CGI Objects demo: VB5CGI object</H2>
<B>CGI script: demoCGI.exe</B>
<HR>
<FORM METHOD="GET" ACTION="http://localhost/scripts/demoCGI.exe">
  Enter a string: <INPUT TYPE="text" NAME="String" SIZE=40 MAXLENGTH=60>
  <INPUT TYPE="submit"  VALUE="submit with GET method">
</FORM>
<FORM METHOD="GET" ACTION="http://localhost/scripts/demoCGI.exe">
  <P><INPUT TYPE="submit"  VALUE="show environment variables">
</FORM>
</BODY>
</HTML>
```

ActiveX DLL

An in-process Dynamic Link Library, which runs in the same process as the client. It provides the fastest way of accessing objects, because property and method calls don't have to be marshaled across process boundaries.

DLL

A Dynamic Link Library is a program module, with functionality for one or multiple programs. When loaded in the memory it can be used simultaneously by multiple threads.

EazyWare
Internet and software solutions, Switzerland

VB5-CGI Objects:
Author: Stephan Schmid
Web: http://www.eazyware.com/vb5-cgi
E-mail: tools@eazyware.com

VB5-CGI Objects

A collection of powerful and easy to use Visual Basic 5 objects to develop CGI scripts on any CGI capable web server under Windows 95/NT4.0.

VB5CGI

An object from the VB5-CGI Objects collection, which handles most of the low-level CGI functionality like standard file input/output.

VB5DB

An object from the VB5-CGI Objects collection, which handles the most used database functionality.

VB5HTML

An object from the VB5-CGI Objects collection, which handles most of the high-level HTML functionality like validating query string parameter values and submitting pages.