

Introduction

The WebComm object can be used to:

- Send data to a web server (GET/POST)
- Retrieve data and files from a web server

The WebComm object is equipped with a few extra's to make your life easier. You can add request headers (for example a "User-Agent" header), retrieve response headers, save files to disk and convert your request to an XML document.

The main object is the HTTP object, which contains most methods and properties. The HTTP object can also contain one or more Header objects. These headers can be request headers or response headers.

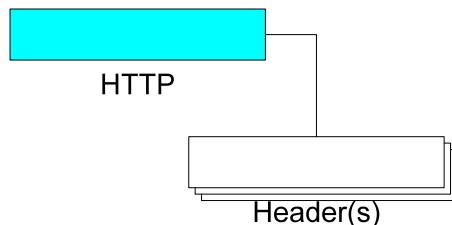
An example application has been included to demonstrate the basic use of the objects. Also, this documentation will demonstrate any property or method with a few lines of code.

Why is this thing useful to me?

It isn't. That's the short answer. The long answer is, I guess; Whenever you need to send data to a web server or retrieve data from a web server you can use this object to do it for you. It is a robust way of sending/retrieving data and has been proven to function properly. It actually is some sort of a wrapper for the MSXML XMLHTTP object.

Documentation:

HTTP Object



Events:

🔧 **ErrorOccured**(lngErrCode As Long, strErrDesc As String)

This event is raised whenever an error occurs. It has two parameters: lngErrCode and strErrDesc which contain information about the error that occurred. These values are also available in the LastErrorNumber and LastErrorDescription properties.

Methods:

 **AddRequestHeader**(strHeaderName As String, strValue As String) As Header


Adds a request header to the request and returns a header object.

Parameters:

strHeaderName: The header to add
strValue : The value for the header

Example:

```
myObject.AddRequestHeader "User-Agent", "WebCommHTTP"
```

 **OpenURL**(strURL As String, [strUser As String], [Optional strPass As String], [Optional vPostBody As Variant]) As Long

OpenURL initiates the HTTP request and returns the HTTP status. Usually this status should be 200 (OK). More about these statuses can be found at:

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10>

Parameters:

strURL : The URL to open.
strUser : *Optional*. If authentication is required, pass the username.
strPass : *Optional*. If authentication is required, pass the password.
vPostBody : *Optional*. Specify this parameter if you want to POST data.

Examples:

```
myObject.OpenURL "http://www.mysite.com"  
myObject.OpenURL "http://www.mysite.com/search.asp?qry=test&id=15"  
myObject.OpenURL "http://www.mysite.com/rss.xml"  
myObject.OpenURL "http://www.mysite.com", "johndoe", "secret"  
myObject.OpenURL "http://www.mysite.com", , , "qry=something"  
myObject.OpenURL "http://www.mysite.com/mylogo.gif"
```

 **SaveToFile**(strFileName As String) As Boolean

Saves the response to a file. Use this method to store binary data or any other response you receiver with OpenURL.

Parameters:

strFile : Filename (with full path) to store data in.

Example:

```
myObject.SaveToFile "C:\mydir\somedata.dat"
```

 **ClearError**()

Clears the LastErrorNumber and LastErrorDescription properties.

Example:

```
myObject.ClearError
```

 **ClearRequestHeaders**()

Clears any set request headers

Example:

```
myObject.ClearRequestHeaders
```

ClearResponseHeaders()

Clears all response headers

Example:

```
myObject.ClearResponseHeaders
```


RemoveRequestHeader(IIndex As Long)


Removes a request header by it's index number.


Example:


```
myObject.RemoveRequestHeader 3
```


Properties:


 **LastErrorDescription()** As String
Read only. Returns the last error description


 **LastErrorNumber()** As Long
Read only. Returns the last error number


 **Raw()** As Variant
Read only. Returns the RAW data received


 **RequestHeader**(IIndex As Long) As Header
Read only. Returns a request header by Index


 **RequestHeaderByName**(strName) As Header
Read only. Returns a request header by header name

 **RequestHeaderCount()** As Long
Read only. Returns the number of request headers added

 **ResponseHeader**(IIndex As Long) As Header
Read only. Returns a response header by Index

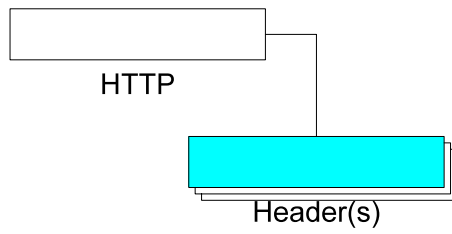
 **ResponseHeaderByName**(strName) As Header
Read only. Returns a response header by header name


 **ResponseHeaderCount()** As Long
Read only. Returns the number of response headers


 **Text()** As String
Read only. Returns the response as a text string

 **XMLDoc()** As Object
Read only. Returns an XML DomDocument object when the received data was valid XML.

Header object



 **HeaderName()** As String
Read/Write. Returns or sets the name of a header (response or request)

 **Value()** As String
Read/Write. Returns or sets the value of a header (response or request)