Exontrol ExHTTP Component

Exontrol's exHTTP is a HTTP client component that provides easy interface to the HTTP (Hypertext Transfer Protocol) protocol. Its primary purpose is to retrieve documents, images from the WEB. With a minimal amount of code you can get a document from the WEB to a string or to a safe array. The exHTTP component doesn't require additional libraries. It just uses the HTTP protocol (rfc2616). For instance, the Image1.Picture = http.GETImage("https://exontrol.com/images/exontrol.png"), assigns the picture from giving URL to Picture property of the Image Object, where the http is a global object of EXHTTPLib.HTTP type.

The exHTTP binary version is 100% free

How to get support?

To keep your business applications running, you need support you can count on.

Here are few hints what to do when you're stuck on your programming:

- Check out the samples - they are here to provide some quick info on how things should be done
- Check out the how-to questions using the eXHelper tool
- Check out the help - includes documentation for each method, property or event
- Check out if you have the latest version, and if you don't have it send an update request here.
- Submit your problem(question) here.

Don't forget that you can contact our development team if you have ideas or requests for new components, by sending us an e-mail at support@exontrol.com (please include the name of the product in the subject, ex: exgrid) . We're sure our team of developers will try to find a way to make you happy - and us too, since we helped.

Regards,
Exontrol Development Team

https://www.exontrol.com

HTTP object

Exontrol's exHTTP is a HTTP client component that provides easy interface to the HTTP (Hypertext Transfer Protocol) protocol. Its primary purpose is to retrieve documents from the WEB. With a minimal amount of code you can get a document from the WEB to a string or to a safe array.

Name	Description
GET	Retrieves the URL to a string or to a safe array of bytes.
GETImage	Retrieves the URL to picture object.
InField	Specifies the value of the field before getting the URL.
OutField	Retrieves the value of the field in the header of the message after getting the URL.
Timeout	Specifies the amount of time (in seconds) the control will wait for the server response.

property HTTP.GET (URL as String) as Variant

Retrieves the URL to a string or to a safe array of bytes.

Type	Description
URL as String	A string expression that indicates the HTTP URL to retrieve. The URL parameter may start with http:// or without. You can specify the web server's port by calling something like: http://www.yourserver.com:8080/index.htm, where, the 8080 is the port to be used. By default, the control uses the port 80.
Variant	A string expression or a safe array of bytes that indicates the body of the message, if the GET operation was ok

Use the GET method to retrieve pages or documents from the web. The GET method waits until the full document is retrieved. Use the Timeout property to specify the amount of time (in seconds) the control will wait for the server response. Use the InField property to add additional fields to the GET request. The InField property has effect only before calling the GET method. The OutField property retrieves the header fields after GET method was performed. The OutField property has effect only after GET method was invoked. The GEM method retrieves a string if the web content is text and if it is not encoded, else a safe array of bytes is retrieved. Use the GETImage property to get a Picture object giving its URL.

The following VB sample displays the **https://www.exontrol.com** web page, in HTML format (before running the following code, you need to add the exHTTP library to the project's references):

```
Dim h As New EXHTTPLib.HTTP
Debug.Print h.GET("https://www.exontrol.com")
```

The following VB sample gets the **https://exontrol.com/images/cart.gif** picture/document to a safe array:

```
Dim h As New EXHTTPLib.HTTP
Dim v As Variant, i
v = h.GET("https://exontrol.com/images/cart.gif")
For Each i In v
   Debug.Print Hex(i)
Next
```

The following VB.NET sample displays the **https://www.exontrol.com** web page (before running the following code, you need to add the exHTTP library to the project's references):

```
Dim n As New EXHTTPLib.HTTP
Debug.WriteLine(n.GET("https://www.exontrol.com").ToString())
```

The following C++ sample displays the **https://www.exontrol.com** web page:

```
#include <atlbase.h>
#import "c:\winnt\system32\ExHTTP.dll"
static COleVariant GET( LPCTSTR szURL, HRESULT* phtError = NULL )
{
   COleVariant vtRequest;
   IUnknown* punkHTTP = NULL;
   OleInitialize( NULL );
   if ( SUCCEEDED( CoCreateInstance( __uuidof( EXHTTPLib::HTTP ), NULL, CLSCTX_ALL, IID_IUnknown, (LPVOID*)&punkHTTP ) ) )
   {
      if ( EXHTTPLib::IHTTPPtr spHTTP( punkHTTP ) )
      {
         USES_CONVERSION;
         HRESULT htError = spHTTP->get_GET( T2OLE(szURL), &vtRequest );
         if ( phtError )
            *phtError = htError;
      }
      punkHTTP->Release();
   }
   OleUninitialize();
   return vtRequest;
}

static CString V2S( VARIANT* pv, LPCTSTR szDefault = _T("") )
{
   if ( pv )
   {
      if ( pv->vt == VT_ERROR )
         return szDefault;
```

```
        COleVariant vt;
        vt.ChangeType( VT_BSTR, pv );
        return V_BSTR( &vt );
    }
    return szDefault;
}
```

property HTTP.GETImage (URL as String) as IPictureDisp

Retrieves the URL to picture object.

Type	Description
URL as String	A string expression that indicates the HTTP URL to retrieve. The URL parameter may start with http:// or without. You can specify the web server's port by calling something like: http://www.yourserver.com:8080/cart.gif, where, the 8080 is the port to be used. By default, the control uses the port 80
IPictureDisp	A Picture object being retrieved.

Use the GETImage method to retrieve pictures from the web. The GETImage method waits until the full picture is retrieved. If no picture document is found at specified URL, the GETImage property fails/throw an error. Use the Timeout property to specify the amount of time (in seconds) the control will wait for the server response. Use the InField property to add additional fields to the GET request. The InField property has effect only before calling the GET method. The OutField property retrieves the header fields after GET method was performed. The OutField property has effect only after GET method was invoked. The GEM method retrieves a string if the web content is text and if it is not encoded, else a safe array of bytes is retrieved. Use the GET method to retrieve pages or documents from the web.

The GETImage can retrieve the following type of pictures:

- The **BMP** file format (*.bmp *.dib *.rle, loads the picture in BMP format), also known as bitmap image file or device independent bitmap (DIB) file format or simply a bitmap, is a raster graphics image file format used to store bitmap digital images, independently of the display device (such as a graphics adapter)
- The **JPEG** file format (*.jpg *.jpe *.jpeg *.jfif, loads the picture in JPEG format, seen most often with the .jpg extension) is a commonly used method of lossy compression for digital images, particularly for those images produced by digital photography.
- The **GIF** (Graphics Interchange Format, *.gif, , loads the picture in GIF format) is a bitmap image format that was introduced by CompuServe in 1987 and has since come into widespread usage on the World Wide Web due to its wide support and portability.
- The **TIFF** (Tagged Image File Format, *.tif *.tiff, loads the picture in TIFF format) is a computer file format for storing raster graphics images, popular among graphic artists, the publishing industry, and both amateur and professional photographers in general.
- The **PNG** (Portable Network Graphics, *.png, loads the picture in PNG format) is a raster graphics file format that supports lossless data compression. PNG was created as an improved, non-patented replacement for Graphics Interchange Format (GIF), and is the most used lossless image compression format on the Internet

- The **EMF** (Enhanced Metafile Format, *.emf, loads the picture in EMF format) is a 32-bit format that can contain both vector information and bitmap information.

For instance, the Image1.Picture = http.GETImage("https://exontrol.com/images/exontrol.png"), assigns the picture from giving URL to Picture property of the Image Object, where the http is a global object of EXHTTPLib.HTTP type.

The following samples shows how can you load pictures using URL (http://), into the eXGrid control.

VBA (MS Access, Excell...)

```
With Grid1
    Set http = CreateObject("Exontrol.HTTP")
    .PictureDisplay = 34
    .Picture = http.GETImage("https://exontrol.com/images/exontrol.png")
End With
```

VB6

```
With Grid1
    Set http = CreateObject("Exontrol.HTTP")
    .PictureDisplay = LowerRight
    .Picture = http.GETImage("https://exontrol.com/images/exontrol.png")
End With
```

VB.NET

```
Dim http
With Exgrid1
    ' Add 'exontrol.exhttp.dll(ExHTTP.dll)' reference to your project.
    http = New exontrol.EXHTTPLib.exhttp()
    .PictureDisplay = exontrol.EXGRIDLib.PictureDisplayEnum.LowerRight
    .Picture = http.get_GETImage("https://exontrol.com/images/exontrol.png")
End With
```

VB.NET for /COM

```
Dim http
With AxGrid1
```

```
    http = CreateObject("Exontrol.HTTP")
    .PictureDisplay = EXGRIDLib.PictureDisplayEnum.LowerRight
    .Picture = http.GETImage("https://exontrol.com/images/exontrol.png")
End With
```

C++

```
/*
    Copy and paste the following directives to your header file as
    it defines the namespace 'EXGRIDLib' for the library: 'ExGrid 1.0 Control Library'

    #import <ExGrid.dll>
    using namespace EXGRIDLib;
*/
EXGRIDLib::IGridPtr spGrid1 = GetDlgItem(IDC_GRID1)->GetControlUnknown();
/*
    Includes the definition for CreateObject function like follows:

    #include <comdef.h>
    IUnknownPtr CreateObject( BSTR Object )
    {
        IUnknownPtr spResult;
        spResult.CreateInstance( Object );
        return spResult;
    };

*/
/*
    Copy and paste the following directives to your header file as
    it defines the namespace 'EXHTTPLib' for the library: 'ExHTTP 1.0 Control Library'

    #import <ExHTTP.dll>
    using namespace EXHTTPLib;
*/
EXHTTPLib::IHTTPPtr http = ::CreateObject(L"Exontrol.HTTP");
spGrid1->PutPictureDisplay(EXGRIDLib::LowerRight);
spGrid1->PutPicture(IPictureDispPtr(((ObjectPtr)(http-
>GetGETImage(L"https://exontrol.com/images/exontrol.png")))));
```

C++ Builder

```
/*
   Select the Component\Import Component...\Import a Type Library,
   to import the following Type Library:

      ExHTTP 1.0 Control Library

   TypeLib: ExHTTP.dll

   to define the namespace: Exhttplib_tlb
*/
//#include "EXHTTPLIB_TLB.h"
Exhttplib_tlb::IHTTPPtr http = Variant::CreateObject(L"Exontrol.HTTP");
Grid1->PictureDisplay = Exgridlib_tlb::PictureDisplayEnum::LowerRight;
Grid1->Picture = (IPictureDisp*)(http-
>get_GETImage(L"https://exontrol.com/images/exontrol.png"));
```

C#

```
// Add 'exontrol.exhttp.dll(ExHTTP.dll)' reference to your project.
// Add 'ExHTTP 1.0 Control Library(ExHTTP.dll)' reference to your project.
exontrol.EXHTTPLib.exhttp http = new exontrol.EXHTTPLib.exhttp();
exgrid1.PictureDisplay = exontrol.EXGRIDLib.PictureDisplayEnum.LowerRight;
exgrid1.Picture = (http.get_GETImage("https://exontrol.com/images/exontrol.png")
as Object);
```

JScript/JavaScript

```
<BODY onload="Init()">
<OBJECT CLASSID="clsid:101EE60F-7B07-48B0-A13A-F32BAE7DA165" id="Grid1">
</OBJECT>
```

```
<SCRIPT LANGUAGE="JScript">
function Init()
{
   var http = new ActiveXObject("Exontrol.HTTP");
   Grid1.PictureDisplay = 34;
   Grid1.Picture = http.GETImage("https://exontrol.com/images/exontrol.png");
}
</SCRIPT>
</BODY>
```

VBScript

```
<BODY onload="Init()">
<OBJECT CLASSID="clsid:101EE60F-7B07-48B0-A13A-F32BAE7DA165" id="Grid1">
</OBJECT>

<SCRIPT LANGUAGE="VBScript">
Function Init()
   With Grid1
      Set http = CreateObject("Exontrol.HTTP")
      .PictureDisplay = 34
      .Picture = http.GETImage("https://exontrol.com/images/exontrol.png")
   End With
End Function
</SCRIPT>
</BODY>
```

C# for /COM

```
// Add 'ExHTTP 1.0 Control Library(ExHTTP.dll)' reference to your project.
EXHTTPLib.HTTP http = new EXHTTPLib.HTTP();
axGrid1.PictureDisplay = EXGRIDLib.PictureDisplayEnum.LowerRight;
(axGrid1.GetOcx() as EXGRIDLib.Grid).Picture =
(http.get_GETImage("https://exontrol.com/images/exontrol.png") as Object);
```

X++ (Dynamics Ax 2009)

```
public void init()
{
    anytype http;
    ;

    super();

    // Add 'exhttp.dll(ExHTTP.dll)' reference to your project.
    // Add 'ExHTTP 1.0 Control Library(ExHTTP.dll)' reference to your project.
    http = COM::createFromObject(new EXHTTPLib.exhttp()); http = http;
    exgrid1.PictureDisplay(34/*LowerRight*/);
    exgrid1.Picture(http.GETImage("https://exontrol.com/images/exontrol.png"));
}
```

Delphi 8 (.NET only)

```
with AxGrid1 do
begin
    http := (ComObj.CreateComObject(ComObj.ProgIDToClassID('Exontrol.HTTP')) as
EXHTTPLib.HTTP);
    PictureDisplay := EXGRIDLib.PictureDisplayEnum.LowerRight;
    (GetOcx() as EXGRIDLib.Grid).Picture :=
(http.GETImage['https://exontrol.com/images/exontrol.png'] as Object);
end
```

Delphi (standard)

```
with Grid1 do
begin
    http :=
(IUnknown(ComObj.CreateComObject(ComObj.ProgIDToClassID('Exontrol.HTTP'))) as
EXHTTPLib_TLB.HTTP);
    PictureDisplay := EXGRIDLib_TLB.LowerRight;
    Picture := (IUnknown(http.GETImage['https://exontrol.com/images/exontrol.png'])
as _TLB.Object);
end
```

VFP

```
with thisform.Grid1
    http = CreateObject("Exontrol.HTTP")
    .PictureDisplay = 34
    .Picture = http.GETImage("https://exontrol.com/images/exontrol.png")
endwith
```

dBASE Plus

```
local http,oGrid

oGrid = form.EXGRIDACTIVEXCONTROL1.nativeObject
http = new OleAutoClient("Exontrol.HTTP")
oGrid.PictureDisplay = 34
oGrid.Picture = http.GETImage("https://exontrol.com/images/exontrol.png")
```

XBasic (Alpha Five)

```
Dim http as P
Dim oGrid as P

oGrid = topparent:CONTROL_ACTIVEX1.activex
http = OLE.Create("Exontrol.HTTP")
oGrid.PictureDisplay = 34
oGrid.Picture = http.GETImage("https://exontrol.com/images/exontrol.png")
```

Visual Objects

```
local http as IHTTP

// Generate Source for 'ExHTTP 1.0 Control Library(ExHTTP.dll)' server from Tools\Automation Server...
http := IHTTP{"Exontrol.HTTP"}
oDCOCX_Exontrol1:PictureDisplay := LowerRight
oDCOCX_Exontrol1:Picture := http:[GETImage,"https://exontrol.com/images/exontrol.png"]
```

PowerBuilder

```
OleObject http,oGrid

oGrid = ole_1.Object
http = CREATE OLEObject
http.ConnectToNewObject("Exontrol.HTTP")
oGrid.PictureDisplay = 34
oGrid.Picture = http.GETImage("https://exontrol.com/images/exontrol.png")
```

Visual DataFlex

```
Procedure OnCreate
    Forward Send OnCreate
    Variant http
    Get Comcreateobject "Exontrol.HTTP" to http
    Set ComPictureDisplay to OLELowerRight
    Set ComPicture to http
End_Procedure
```

XBase++

```
#include "AppEvent.ch"
#include "ActiveX.ch"

PROCEDURE Main
    LOCAL oForm
    LOCAL nEvent := 0, mp1 := NIL, mp2 := NIL, oXbp := NIL
    LOCAL oGrid
    LOCAL http

    oForm := XbpDialog():new( AppDesktop() )
    oForm:drawingArea:clipChildren := .T.
    oForm:create( ,,{100,100}, {640,480},, .F. )
    oForm:close  := {|| PostAppEvent( xbeP_Quit )}
    oGrid := XbpActiveXControl():new( oForm:drawingArea )
    oGrid:CLSID  := "Exontrol.Grid.1" /*{101EE60F-7B07-48B0-A13A-F32BAE7DA165}*/
    oGrid:create(,, {10,60},{610,370} )

        http := CreateObject("Exontrol.HTTP")
        oGrid:PictureDisplay := 34/*LowerRight*/
        oGrid:Picture := http:GETImage("https://exontrol.com/images/exontrol.png")

    oForm:Show()
    DO WHILE nEvent != xbeP_Quit
        nEvent := AppEvent( @mp1, @mp2, @oXbp )
        oXbp:handleEvent( nEvent, mp1, mp2 )
    ENDDO
RETURN
```

property HTTP.InField(FieldName as String) as String

Specifies the value of the field before getting the URL.

Type	Description
FieldName as String	A String expression that indicates the name of the field being added to the GET header, before sending the request.
String	A String expression that indicates the value of the field.

The InField property retrieves or specifies the "request" fields. If the FieldName parameter is empty, the InField property retrieves all the fields in the GET header. Use the InField property to add addition fields to the GET request. The RFC 2616, explains better all the fields that may be used. The OutField property lists the repose fields. Use the GET method to retrieve a document or a web page.

Shortly here they are:

- **Accept**, The Accept request-header field can be used to specify certain media types which are acceptable for the response. By default, the **Accept** field is "*/*"
- Accept-Charset**,** The Accept-Charset request-header field can be used to indicate what character sets are acceptable for the response. By default, the Accept-Charset field is not used.
- Accept-Encoding, The Accept-Encoding request-header field is similar to Accept, but restricts the content-codings that are acceptable in the response. By default, the Accept-Encoding field is not used**.**
- **Accept-Language**, The Accept-Language request-header field is similar to Accept, but restricts the set of natural languages that are preferred as a response to the request. Language tags. By default, the **Accept-Language** field is "en-us".
- Authorization, A user agent that wishes to authenticate itself with a server-- usually, but not necessarily, after receiving a 401 response--does so by including an Authorization request-header field with the request. By default, this field is not used.
- Expect, The Expect request-header field is used to indicate that particular server behaviors are required by the client, By default, this field is not used.
- From, The From request-header field, if given, SHOULD contain an Internet e-mail address for the human user who controls the requesting user agent. By default, this field is not used
- Host**.** The Host request-header field specifies the Internet host and port number of the resource being requested, as obtained from the original URI given by the user or referring resource (generally an HTTP URL). This field is automatically get from the URL.
- If-Match, The If-Match request-header field is used with a method to make it conditional. A client that has one or more entities previously obtained from the resource

can verify that one of those entities is current by including a list of their associated entity tags in the If-Match header field. By default, this field is not used.

- If-Modified-Since, The If-Modified-Since request-header field is used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, an entity will not be returned from the server; instead, a 304 (not modified) response will be returned without any message-body. This field is not used.
- If-None-Match, The If-None-Match request-header field is used with a method to make it conditional. A client that has one or more entities previously obtained from the resource can verify that none of those entities is current by including a list of their associated entity tags in the If-None-Match header field. By default, this field is not used.
- If-Range, If a client has a partial copy of an entity in its cache, and wishes to have an up-to-date copy of the entire entity in its cache, it could use the Range request-header with a conditional GET (using either or both of If-Unmodified-Since and If-Match.) However, if the condition fails because the entity has been modified, the client would then have to make a second request to obtain the entire current entity-body. By default, this field is not used.
- If-Unmodified-Since, The If-Unmodified-Since request-header field is used with a method to make it conditional. If the requested resource has not been modified since the time specified in this field, the server SHOULD perform the requested operation as if the If-Unmodified-Since header were not present. By default, this field is not used.
- Proxy-Authorization, The Proxy-Authorization request-header field allows the client to identify itself (or its user) to a proxy which requires authentication. The Proxy-Authorization field value consists of credentials containing the authentication information of the user agent for the proxy and/or realm of the resource being requested. By default, this field is not used.
- Referer, The Referer[sic] request-header field allows the client to specify, for the server's benefit, the address (URI) of the resource from which the Request-URI was obtained (the "referrer", although the header field is misspelled.) The Referer request-header allows a server to generate lists of back-links to resources for interest, logging, optimized caching, etc. This field can not be changed.
- TE, The TE request-header field indicates what extension transfer-codings it is willing to accept in the response and whether or not it is willing to accept trailer fields in a chunked transfer-coding. By default, this field is not used.
- **User-Agent**, The User-Agent request-header field contains information about the user agent originating the request. This is for statistical purposes, the tracing of protocol violations, and automated recognition of user agents for the sake of tailoring responses to avoid particular user agent limitations. By default, this field is "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)"
- Vary, The Vary field value indicates the set of request-header fields that fully determines, while the response is fresh, whether a cache is permitted to use the response to reply to a subsequent request without revalidation. By default, this field is not used.

property HTTP.OutField (URL as String, FieldName as String) as String

Retrieves the value of the field in the header of the message after getting the URL.

Type	Description
URL as String	A string expression that indicates the HTTP URL to retrieve. The URL parameter may start with http:// or without. You can specify the web server's port by calling something like: http://www.yourserver.com:8080/index.htm, where, the 8080 is the port to be used. By default, the control uses the port 80.
FieldName as String	A String expression that indicates the name of the response field. If the FieldName property is empty, the OutField property retrieves all the response fields, line by line. Lines are separated by crlf sequence (/n/r characters)
String	A String expression that indicates the value of the field.

The OutField property retrieves a particular response field or all fields. For instance, the OutField("www.exontrol.com","") retrieves all fields in the header. Use the GET method to retrieve the document or a web page. Use the InField property to specify request fields before GET or OutField method is called. Use the Timeout property to specify the amount of time (in seconds) the control will wait for the server response.

The OutField property includes all response fields line by line. For instance, the OutField("www.exontrol.com","Content-Type") retrieves the type of the document being requested.

The FieldName parameter may be one of the following:

- If the FieldName parameter is empty, the OutField retrieves all the response fields
- Accept-Ranges, The Accept-Ranges response-header field allows the server to indicate its acceptance of range requests for a resource
- Age, The Age response-header field conveys the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.
- ETag, The ETag response-header field provides the current value of the entity tag for the requested variant
- **Location**, The Location response-header field is used to redirect the recipient to a location other than the Request-URI for completion of the request or identification of a new resource.
- Proxy-Authenticate, the field value consists of a challenge that indicates the authentication scheme and parameters applicable to the proxy for this Request-URI

- Retry-After, The Retry-After response-header field can be used with a 503 (Service Unavailable) response to indicate how long the service is expected to be unavailable to the requesting client.
- Server, The Server response-header field contains information about the software used by the origin server to handle the request
- WWW-Authenticate, The field value consists of at least one challenge that indicates the authentication scheme(s) and parameters applicable to the Request-URI

The OutField property may include other response fields, based on the web server is queering.

property HTTP.Timeout as Long

Specifies the amount of time (in seconds) the control will wait for the server response.

Type	Description
Long	A long expression that specifies the amount of time (in seconds) the control will wait for the server response.

By default, the Timeout property is 5 seconds