



## Help for Gopher

[Properties](#)

[Events](#)

[Methods](#)

[Frequently Asked Questions](#)

### Registration Information

### Order Form

### Getting Custom Controls Written

### Licensing Information

#### **Description**

Gopher is the predecessor to the World Wide Web. Universities still have a lot of information out on Gopher sites. The Gopher control allows your program to automatically retrieve information, menus, images, etc. from Gopher sites. The Gopher control even gives you access to Veronica, which allows you to search Gopher-space.

#### **File Name**

GOPHER1.VBX, GOPHER32.OCX

#### **ActiveX / OCX Object Name**

Mabry.GopherCtrl

#### **ActiveX Compatibility**

VB 4.0 (32-bit) and 5.0

#### **ActiveX Built With**

Microsoft Visual C++ v4

#### **ActiveX - Required DLLs**

MFC40.DLL (October 6th, 1995 or later)

OLEPRO32.DLL (October 6th, 1995 or later)

MSVCRT40.DLL (September 29th, 1995 or later)

#### **VBX Object Type**

Gopher

#### **VBX Compatibility**

VB 2.0, 3.0 and 4.0 (16-bit)

#### **VBX Built With**

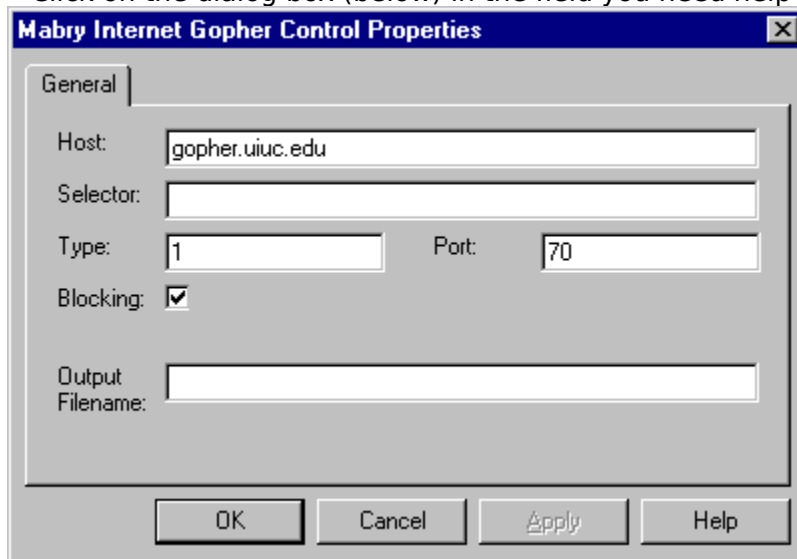
Microsoft Visual C++ v1.5

**Distribution Note** When you develop and distribute an application that uses this control, you should install the control file into the user's Windows SYSTEM directory. The control file has version information built into it. So, during installation, you should ensure that you are not overwriting a newer version.

---

## Gopher Property Page

Click on the dialog box (below) in the field you need help for.



The image shows a Windows-style dialog box titled "Mabry Internet Gopher Control Properties". The "General" tab is selected. The dialog contains the following fields and controls:

- Host:** A text box containing "gopher.uiuc.edu".
- Selector:** An empty text box.
- Type:** A dropdown menu showing "1".
- Port:** A text box containing "70".
- Blocking:** A checked checkbox.
- Output Filename:** An empty text box.

At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

Close

## Gopher Properties

Properties that have special meaning for this control or that only apply to this control are marked with an asterisk (\*).

- \***Action** Property
- \***Blocking** Property
- \***Host** Property
- \***ItemCount** Property
- \***ItemDescription** Property
- \***ItemExtra** Property
- \***ItemHost** Property
- \***ItemPort** Property
- \***ItemSelector** Property
- \***ItemType** Property
- \***LastError** Property
- \***OutputFilename** Property
- \***Port** Property
- \***Selector** Property
- \***Type** Property
- \***Version** Property

Close

## Gopher Events

Events that have special meaning for this control or that only apply to this control are marked with an asterisk (\*).

\***AsyncError** Event

\***Done** Event

\***OnDownload** Event

Close

## Gopher Methods

Methods that have special meaning for this control or that only apply to this control are marked with an asterisk (\*).

\*AboutBox Method

\*GetGopher Method

Close

## Frequently Asked Questions

### Internet Pack - General Questions

[Why won't my Internet Pack VBXes load into VB?](#)

[Why do I get a GPF when I try to unload my form \(or control\) from the Done event?](#)

[With which TCP/IP stacks have your Internet controls been tested?](#)

[How do I enable/disable the Windows 95 Dial-Up Networking connect prompt when my application issues a Connect method?](#)

[Why won't my Internet Pack VBX load?](#)

[How do I convert my code from BLOCKING \(Synchronous\) to NON-BLOCKING \(Asynchronous\)?](#)

[How can I detect whether someone has entered an IP or host name?](#)

[What is the meaning of Error 20002 "unexpected server response"?](#)

[Can you recommend any good books that will help me understand Internet programming better?](#)

### Internet Pack - Blocking

[I'm unclear on blocking. Can you explain it to me?](#)

[Should I use blocking or non-blocking calls?](#)

[Why do I keep getting the error "Busy executing asynchronous command"?](#)

[Why do I keep getting errors when using an Internet VBX control?](#)

### Internet Pack - Debugging

[How do I tell what's happening when your control is talking to a server?](#)

## **Why won't my Internet Pack VBXes load into VB?**

### Frequently Asked Questions

The VBXes are looking for a file called WINSOCK.DLL. This DLL should be in your Windows directory (most DLLs are located in your Windows\System directory -- this one is an exception). Look for WINSOCK.DLL. If it's not in your Windows directory, we recommend moving it there. Be sure to write down where it was, in case something goes wrong.

Also, check the date on your WINSOCK.DLL. If it's 1994 or before, you should look into getting a later version.

## **Why do I get a GPF when I try to unload my form (or control) from the Done event?**

### Frequently Asked Questions

This is not uncommon in many controls. If the form containing the control is unloaded but the control's C++ code for the event has to reference the control, the GPF will occur because the control is no longer available after it is has been unloaded. The solution is to enable a timer in the Done event and have the Timer unload the form (or control).



## **With which TCP/IP stacks have your Internet controls been tested?**

### Frequently Asked Questions

The majority of our internal testing is done on either NT's or Win95's standard stacks. We also utilize a 3.1 machine running Trumpet Winsock.

As part of our beta program, the controls wind up on a variety of stacks like Novell (known to have differences in Winsock, but should be OK with the latest patches from Novell), WFWG (also has a known problem that can cause FTP trouble, but MS has a patch for that product as well (article ID Q122544)).

The controls support the standard Winsock interface, so in general, the 16-bit environments that do not come with a default stack (i.e., Windows 3.x) may involve a bit more setup, but as long as some reputable stack is used, there shouldn't be any problems.

## **How do I enable/disable the Windows 95 Dial-Up Networking connect prompt when my application issues a Connect method?**

### Frequently Asked Questions

The fact that the DUN pops up when attempting to establish a network connection is a Win95 OS setting. To change this behavior, choose Dial Up Networking from "My Computer", and select "Settings..." from the "Connections" menu. Set the desired value in the "When establishing a network connection" frame.

## **Why won't my Internet Pack VBX load?**

### Frequently Asked Questions

Usually, the Internet VBXes won't load when the WINSOCK.DLL is missing. Make sure you have a current WINSOCK.DLL in the Windows or Windows\System sub-dir. Some versions of Windows 3.x WinSocks may actually require a TCP/IP connection.

## How do I convert my code from BLOCKING (Synchronous) to NON-BLOCKING (Asynchronous)?

### Frequently Asked Questions

A quick fix for converting Blocking code to non-blocking code is as follows:

```
Blocking=False
```

In the Declarations of the Form, add:

```
Private fDone as Boolean
```

In the Done event of the control set the fDone flag as shown:

```
Private Sub FTP1_Done()  
    fDone = True  
End Sub
```

Then, when invoking a method, just loop until the Done event sets the fDone flag.

```
fDone = False  
mMail1.Connect  
Do  
    DoEvents  
    'here is where your application  
    'can do other things  
Loop Until (fDone)
```

Note: you may want to set a timer in the loop so it will not loop endlessly should some problem occur. Also, depending upon your code you may want to conditionally set the fDone flag in the AsyncError event.

## How can I detect whether someone has entered an IP or host name?

### Frequently Asked Questions

You can use the function (below) to check for a host name or IP address.

```
' This Function receives a string argument and
' validates whether the string is a valid IP value,
' by verifying that it is in the format of w.x.y.z and
' that each octet is between 0 and 255
,
' Returns True if IP there are 4 octets and each is
' between 0 and 255.
,
' Returns False in all other cases
,
' Disclaimer -- this function will not detect certain
' values such as netmasks like 255.255.255.255,
' which meet the criteria but are not valid IPs.
,
```

```
Private Function Valid_IP(IP As String) As Boolean
    Dim i As Integer
    Dim dot_count As Integer
    Dim test_octet As String

    IP = Trim$(IP)

    ' make sure the IP long enough before
    ' continuing
    If Len(IP) < 8 Then
        Valid_IP = False
        Exit Function
    End If

    i = 1
    dot_count = 0
    For i = 1 To Len(IP)
        If Mid$(IP, i, 1) = "." Then
            ' increment the dot count and
            ' clear the test octet variable
            dot_count = dot_count + 1
            test_octet = ""
            If i = Len(IP) Then
                ' we've ended with a dot
                ' this is not good
                Valid_IP = False
                Exit Function
            End If
        Else
            test_octet = test_octet & Mid$(IP, i, 1)
            On Error Resume Next
            byte_check = CByte(test_octet)
            If (Err) Then
                ' either the value is not numeric
                ' or exceeds the range of the byte
                ' data type.
                Valid_IP = False
                Exit Function
            End If
        End If
    Next i
    Valid_IP = (dot_count = 3)
End Function
```

```
        End If
    End If
Next i
' so far, so good
' did we get the correct number of dots?
If dot_count <> 3 Then
    Valid_IP = False
    Exit Function
End If
' we have a valid IP format!
Valid_IP = True
End Function
```

## **What is the meaning of Error 20002 "unexpected server response"?**

### Frequently Asked Questions

The control has issued some command and the server did not accept it. It could be anything from an improperly formatted e-mail address to an unimplemented command on the server. You'll have to enable debugging to see what the command and reply are.

## **Can you recommend any good books that will help me understand Internet programming better?**

### Frequently Asked Questions

Any good book on TCP/IP would be helpful. From personal experience, tech support recommends "TCP/IP" by Dr. Sidnie Feit, published by McGraw-Hill . It is *not* written from a programming standpoint, but does include everything you'd want to know about the lower levels of the OSI stack (including TCP/UDP/IP, etc.).



## **I'm unclear on blocking. Can you explain it to me?**

### Frequently Asked Questions

When your application requests data from a network connection, it is hard to predict how long it will take before the data arrives and the call can complete. As a programmer, you have to determine whether to wait for the outcome of the call, or return immediately to your application and get the data *when* the data arrives.

Calls that wait, are called blocking calls. Because the call must complete before the application continues, blocking calls are also referred to as synchronous calls.

Calls that return control to your application immediately are called non-blocking calls. Since your application can perform tasks while the call is retrieving the data, non-blocking calls are also referred to as asynchronous calls.

Mabry Internet controls support both blocking and non-blocking calls.

It is important to note that even when using blocking calls, Windows can send event messages (such as Timer events, mouse clicks, etc.) to your application and it can respond to them. This can result in errors. It is the responsibility of the programmer to minimize the likelihood of these situations (such as disabling any Timers or command buttons that will interrupt the call) and handle any errors should such conditions arise.

Error handling is very important when issuing calls to a network. Always use some method of On Error handling when invoking blocking calls. For non-blocking calls, normal On Error handling is required in addition to responding to the AsyncError event.

## **Should I use blocking or non-blocking calls?**

Frequently Asked Questions

It depends on your application. See the explanation on blocking calls for a complete description of blocking vs. non-blocking.

## **Why do I keep getting the error "Busy executing asynchronous command"?**

### Frequently Asked Questions

A call has been invoked but a previous call has not been completed yet. Either set Blocking mode to true or wait for the Done event before issuing subsequent commands.

It is important to note that even when using blocking calls, Windows can send event messages (such as Timer events, mouse clicks, etc.) to your application and it can respond to them. This can result in errors. It is the responsibility of the programmer to minimize the likelihood of these situations (such as disabling any Timers or command buttons that will interrupt the call) and handle any errors should such conditions arise.

## **Why do I keep getting errors when using an Internet VBX control?**

### Frequently Asked Questions

A call has been invoked but a previous call has not been completed yet. Either set Blocking mode to true or wait for the Done event before issuing subsequent commands.

It is important to note that even when using blocking calls, Windows can send event messages (such as Timer events, mouse clicks, etc.) to your application and it can respond to them. This can result in errors. It is the responsibility of the programmer to minimize the likelihood of these situations (such as disabling any Timers or command buttons that will interrupt the call) and handle any errors should such conditions arise.

## **How do I tell what's happening when your control is talking to a server?**

### Frequently Asked Questions

The Internet Pack controls have debugging support built-in. Simply set the Debug property on the control to 1 and then add the following code to the Debug event of the control:

```
Debug.Print Message
```

## Registration Information

### CREDITS

Gopher was written by Zane Thomas.

### CONTACT INFORMATION

Orders, inquiries, technical support, questions, comments, etc. can be sent to [mabry@mabry.com](mailto:mabry@mabry.com) on the Internet. Our mailing address/contact information is:

Mabry Software, Inc.  
Post Office Box 31926  
Seattle, WA 98103-1926  
Sales: 1-800-99-MABRY (U.S. Only)  
Voice: 206-634-1443  
Fax: 206-632-0272 or 206-364-3196  
Web: <http://www.mabry.com>

### COST

The price of Gopher (control only) is US\$35 (US\$40 for International orders). The cost of Gopher and the C/C++ source code (of the control itself) is US\$90 (US\$95 for International orders).

Prices are subject to change without notice.

### DELIVERY METHODS

We can ship this software to you via air mail and/or e-mail.

**Air Mail** - you will receive disks, a printed manual, and printed receipt if you choose this delivery method. The costs are:

US\$5.00	US Priority Mail
US\$10.00	AirBorne Express 2nd Day (US deliveries only)
US\$15.00	AirBorne Express Overnight (US deliveries only)
US\$45.00	International AirBorne Express.

**E-Mail** - We can ship this package to you via e-mail. You need to have an e-mail account that can accept large file attachments (which includes CompuServe, AOL, and most Internet providers). If you choose this option, please note: a printed manual is not included. We will, however, e-mail a receipt to you.

Be sure to include your full mailing address with your order. Sometimes (on the Internet) the package cannot be e-mailed, so we are forced to send it through the normal mails.

**CompuServe E-Mail** - CompuServe members can use the software registration forum (GO SWREG) to register this package. Gopher's SWREG ID number is 6394. The source code version's ID number is 9061. PLEASE NOTE: When you order through SWREG, we send the registered package to your CompuServe account (not your Internet or AOL account) within a few hours.

### ORDER / PAYMENT METHODS

You can order this software by phone, fax, e-mail, mail. For your convenience, an order form has been provided that you can print out directly from this help file.

Please note that orders must include all information that is requested on our order form. Your shipment WILL BE DELAYED if we have to contact you for additional information (such as phone number, street address, etc.).

You can pay by credit card (VISA, MasterCard, American Express), check (U.S. dollars drawn on a U.S. bank), cash, International Money Order, International Postal Order, Purchase Order (established business entities only - terms net 30), or wire transfer.

## **WIRE TRANSFER INFORMATION**

Here is the information you need regarding our account for a wire funds transfer:

Bank Name: SeaFirst - Stone Way Branch  
Bank Address: 3601 Stone Way North  
Seattle, WA 98103  
Bank Phone: 206-585-4951  
Account Name: Mabry Software, Inc.  
Routing Number: 12000024  
Account Number: 16311706

If you are paying with a wire transfer of funds, please add US\$12.50 to your order. This is the fee that SeaFirst Bank charges Mabry Software. Also, please ADD ANY ADDITIONAL FEES THAT YOUR BANK MAY CHARGE for wire transfer service. If you are paying with a wire transfer, we must have full payment deposited to our account before we can ship your order.

Copyright © 1996-1997 by Mabry Software, Inc.



# Gopher Order Form

Use the Print Topic... command from the File menu to print this order form.

**Mail this form to:** Mabry Software, Inc.  
Post Office Box 31926  
Seattle, WA 98103-1926  
Phone: 206-634-1443  
Fax: 206-632-0272 or 206-364-3196  
Internet: mabry@mabry.com  
Web: www.mabry.com

Where did you get this copy of Gopher?

\_\_\_\_\_

Name: \_\_\_\_\_

Ship to: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Phone: \_\_\_\_\_

Fax: \_\_\_\_\_

E-Mail: \_\_\_\_\_

MC/VISA/AMEX: \_\_\_\_\_ exp. \_\_\_\_\_

P.O. # (if any): \_\_\_\_\_ Signature \_\_\_\_\_

qty ordered \_\_\_\_\_ REGISTRATION  
\$35.00 (\$40.00 international). Check or money order in U.S.  
currency drawn on a U.S. bank. Add \$5.00 per order for shipping  
and handling.

qty ordered \_\_\_\_\_ SOURCE CODE AND REGISTRATION  
\$90.00 (\$95.00 international). Check or money order in U.S.  
currency drawn on a U.S. bank. Add \$5.00 per order for shipping  
and handling.



## Error Codes

<b>Constant</b>	<b>Value</b>	<b>Description</b>
	0	No error.
WSAEINTR	10004	System level interrupt interrupted socket operation.
WSAEBADF	10009	Generic error for invalid format, bad format.
WSAEACCES	10013	Generic error for access violation.
WSAEFAULT	10014	Generic error for fault.
WSAEINVAL	10022	Generic error for invalid format, entry, etc.
WSAEMFILE	10024	Generic error for file error.
	10025	The IP address provided is not valid or the host specified by the IP does not exist.
WSAENOTSOCK	10038	Invalid socket or not connected to remote.
WSAEADDRINUSE	10048	The specified address is already in use.
WSAEADDRNOTAVAIL	10049	The specified address is not available.
WSAENETDOWN	10050	The connected network is not available.
WSAENETUNREACH	10051	The connected network is not reachable.
WSAENETRESET	10052	The connected network connection has been reset.
WSAECONNABORTED	10053	The current connection has been aborted by the network or intermediate services.
WSAECONNRESET	10054	The current socket connection has been reset.
WSAENOTCONN	10057	The current socket has not been connected.
WSAESHUTDOWN	10058	The connection has been shutdown.
WSAETIMEDOUT	10060	The current connection has timedout.
WSAECONNREFUSED	10061	The requested connection has been refused by the remote host.
WSAENAMETOOLONG	10063	Specified host name is too long.
WSAEHOSTDOWN	10064	Remote host is currently unavailable.
WSAEHOSTUNREACH	10065	Remote host is currently unreachable.
WSASYSNOTREADY	10091	Remote system is not ready.
WSAVERNOTSUPPORTED	10092	Current socket version not supported by application.
WSANOTINITIALISED	10093	Socket API is not initialized.
WSAEDISCON	10101	Socket has been disconnected.

## See Also

[Done](#) Event

[GetGopher](#) Method

[Host](#) Property

[Port](#) Property

[Selector](#) Property

[Type](#) Property

**See Also**

**Action** Property

**Done** Event

**GetGopher** Method

## See Also

[Action Property](#)

[AsyncError Event](#)

[Done Event](#)

[GetGopher Method](#)

## See Also

[\*\*Action\*\* Property](#)

[\*\*AsyncError\*\* Event](#)

[\*\*Blocking\*\* Property](#)

[\*\*GetGopher\*\* Method](#)

[\*\*ItemCount\*\* Property](#)

[\*\*ItemDescription\*\* Property](#)

[\*\*ItemExtra\*\* Property](#)

[\*\*ItemHost\*\* Property](#)

[\*\*ItemPort\*\* Property](#)

[\*\*ItemType\*\* Property](#)

## See Also

[\*\*Action\*\* Property](#)

[\*\*AsyncError\*\* Event](#)

[\*\*Blocking\*\* Property](#)

[\*\*Done\*\* Event](#)

[\*\*Host\*\* Property](#)

[\*\*ItemCount\*\* Property](#)

[\*\*ItemDescription\*\* Property](#)

[\*\*ItemExtra\*\* Property](#)

[\*\*ItemHost\*\* Property](#)

[\*\*ItemPort\*\* Property](#)

[\*\*ItemSelector\*\* Property](#)

[\*\*ItemType\*\* Property](#)

[\*\*OutputFilename\*\* Property](#)

[\*\*Port\*\* Property](#)

[\*\*Selector\*\* Property](#)

[\*\*Type\*\* Property](#)

## See Also

[\*\*Action\*\* Property](#)

[\*\*Blocking\*\* Property](#)

[\*\*Done\*\* Event](#)

[\*\*GetGopher\*\* Method](#)

[\*\*Port\*\* Property](#)

## See Also

[\*\*Action\*\* Property](#)

[\*\*Blocking\*\* Property](#)

[\*\*Done\*\* Event](#)

[\*\*GetGopher\*\* Method](#)

[\*\*ItemDescription\*\* Property](#)

[\*\*ItemExtra\*\* Property](#)

[\*\*ItemHost\*\* Property](#)

[\*\*ItemPort\*\* Property](#)

[\*\*ItemSelector\*\* Property](#)

[\*\*ItemType\*\* Property](#)



## See Also

[\*\*Action\*\* Property](#)

[\*\*Done\*\* Event](#)

[\*\*GetGopher\*\* Method](#)

[\*\*Host\*\* Property](#)

[\*\*Port\*\* Property](#)

[\*\*Selector\*\* Property](#)

[\*\*Type\*\* Property](#)

## See Also

[\*\*Action\*\* Property](#)

[\*\*Done\*\* Event](#)

[\*\*GetGopher\*\* Method](#)

[\*\*Host\*\* Property](#)

[\*\*Port\*\* Property](#)

[\*\*Selector\*\* Property](#)

[\*\*Type\*\* Property](#)

## See Also

[\*\*Action\*\* Property](#)

[\*\*Done\*\* Event](#)

[\*\*GetGopher\*\* Method](#)

[\*\*Host\*\* Property](#)

[\*\*Port\*\* Property](#)

[\*\*Selector\*\* Property](#)

[\*\*Type\*\* Property](#)

## See Also

[\*\*Action\*\* Property](#)

[\*\*Done\*\* Event](#)

[\*\*GetGopher\*\* Method](#)

[\*\*Host\*\* Property](#)

[\*\*Port\*\* Property](#)

[\*\*Selector\*\* Property](#)

[\*\*Type\*\* Property](#)

## See Also

[\*\*Action\*\* Property](#)

[\*\*Done\*\* Event](#)

[\*\*GetGopher\*\* Method](#)

[\*\*Host\*\* Property](#)

[\*\*Port\*\* Property](#)

[\*\*Selector\*\* Property](#)

[\*\*Type\*\* Property](#)

## See Also

[\*\*Action\*\* Property](#)

[\*\*Done\*\* Event](#)

[\*\*GetGopher\*\* Method](#)

[\*\*Host\*\* Property](#)

[\*\*Port\*\* Property](#)

[\*\*Selector\*\* Property](#)

[\*\*Type\*\* Property](#)

## See Also

[Action Property](#)

[AsyncError Event](#)

[Done Event](#)

## See Also

[Action](#) Property

[Done](#) Event

[GetGopher](#) Method

[OutputFilename](#) Property



## See Also

[Action Property](#)

[GetGopher Method](#)

[Host Property](#)

[Port Property](#)

[Selector Property](#)

## See Also

[Action Property](#)

[GetGopher Method](#)

[Host Property](#)

## See Also

[Action Property](#)

[GetGopher Method](#)

[Host Property](#)

[Port Property](#)

[Type Property](#)

## See Also

[Action Property](#)

[GetGopher Method](#)

[Host Property](#)

[Port Property](#)

[Selector Property](#)

## AboutBox Method

[Error Codes](#)

### Description

displays the About Box for the control.

### Syntax

*object*.**AboutBox**

The syntax of the **AboutBox** method has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	Required. A Gopher control.

### Remarks

This method displays the About Box for this control which includes copyright information.

## Action Property

[See Also](#)

[Error Codes](#)

### Description

Causes control to initiate a command / method.

### Syntax

*object.Action* [= *action* ]

The syntax of the **Action** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
--------------------	---------------------------

<i>object</i>	A Gopher control.
---------------	-------------------

<i>action</i>	An integer expression that determines the action performed by the control.
---------------	--

### Remarks

Setting this property makes the Gopher control perform an action. The action depends on the value set. Gopher accepts the following:

<b><u>Value</u></b>	<b><u>Description</u></b>
---------------------	---------------------------

1	Retrieves the information (if any) from the site specified. Typically, this is a Gopher menu. Sometimes, this is a binary file, image file or text file. The type of retrieval depends on the <a href="#">Type property</a> . See the <a href="#">GetGopher method</a> for more information.
---	--

### Data Type

Integer (enumerated)

## AsyncError Event

[See Also](#)

[Error Codes](#)

### Description

Fired when an error occurs during asynchronous operations

### Syntax

**Sub** *object* **AsyncError**([*index* **As Integer**,] *errornumber* **As Integer**, *errormessage* **As String**)

The syntax of the **AsyncError** event has these parts:

<b>Part</b>	<b>Description</b>
<i>object</i>	A Gopher control.
<i>index</i>	An integer that identifies a control if it's in a control array.
<i>errornumber</i>	An integer that holds the current error number.
<i>errormessage</i>	A string expression that holds text describing the current error.

### Remarks

If an error occurs during the execution of asynchronous commands (only possible when [Blocking](#) is set to False) the program is notified by firing the AsyncError event.

## Blocking Property

[See Also](#)

[Error Codes](#)

### Description

Determines if any methods or actions are blocking (synchronous).

### Syntax

*object*.**Blocking** [= *boolean* ]

The syntax of the **Blocking** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Gopher control.
<i>boolean</i>	A boolean flag that determines if the control waits until a command is finished, or, returns control immediately and then fires an event when done.

### Remarks

If this property is set to True, any commands using either the [Action property](#) or the [GetGopher method](#) will not return to your code until the command completes. In other words, the command will be handled synchronously.

If this property is False, any commands are handled asynchronously. They return to you immediately. You are notified of completion with the [Done event](#).

### Data Type

Integer (boolean)



## Done Event

[See Also](#)

[Error Codes](#)

### Description

This event procedure is fired when the control completes an action or method.

### Syntax

**Sub** *object\_Done*([*index As Integer*,] *errornumber As Integer*, *errornumber As Integer*)

The syntax of the **Done** event has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Gopher control.
<i>index</i>	An integer that identifies a control if it's in a control array.
<i>errornumber</i>	An integer that holds the current error number.
<i>errornumber</i>	An integer that specifies an error code. Zero means no error.

### Remarks

This event fires when the data request completes. If the *ErrorNumber* parameter is zero, the Gopher data is either in the *ItemXXX* properties or saved in a file on the disk (see [OutputFilename](#)). The destination depends on the [Type property](#).

## GetGopher Method

[See Also](#)

[Error Codes](#)

### Description

Gets user information from specified server.

### Syntax

*object*.**GetGopher**

The syntax of the **GetGopher** method has these parts:

<b>Part</b>	<b>Description</b>
<i>object</i>	Required. A Gopher control.

### Remarks

This method retrieves a Gopher item from the [Host](#), [Selector](#), [Port](#) and [Type](#) combination specified. [Host](#) determines which computer to access, [Port](#) tells which host port to access, [Selector](#) determines what directory or filename to access, and [Type](#) determines what kind of data to retrieve.

Sometimes the [OutputFilename property](#) is required to retrieve data. This is only used when the Gopher item to retrieve is a file of some kind. See the [Type](#) property for more information.

This method can be called in one of two ways: blocking or non-blocking. This is determined by the *blocking* parameter or, if the *blocking* parameter is not specified, the [Blocking property](#). If blocking, the Gopher control does not return control to your program until it retrieves the queried data from the host specified or until it times-out.

With non-blocking the control returns immediately. The control fires the [Done event](#) upon completion (good or bad). Sometimes, with the non-blocking method, an error is reported (10035) immediately. This is only a warning from the WinSock layer letting you know that the connection to the host is busy sending data and requires a little time to send your request. It can be treated as normal.

## Host Property

[See Also](#)

[Error Codes](#)

### Description

Gopher requests information from the host with this address.

### Syntax

*object*.**Host** [= *host* ]

The syntax of the **Host** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
--------------------	---------------------------

<i>object</i>	A Gopher control.
---------------	-------------------

<i>host</i>	A string expression that specifies the name or IP address of a Gopher server.
-------------	---

### Remarks

This property determines which host Gopher will query.

### Data Type

String

## ItemCount Property

[See Also](#)

[Error Codes](#)

### Description

Holds the number of menu items retrieved.

### Syntax

*object*.**ItemCount**

The syntax of the **ItemCount** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Gopher control.

### Remarks

ItemCount holds the number of menu items retrieved from the Gopher site. This property tells you how many entries are in the [ItemHost](#), [ItemPort](#), [ItemType](#), [ItemDescription](#), [ItemSelector](#) and [ItemExtra](#) property arrays.

This property is only valid after searching using the [Action property](#) or [GetGopher method](#). This property is read-only and only available at run-time.

### Data Type

Integer

## ItemDescription Property

[See Also](#)

[Error Codes](#)

### Description

Holds the human-readable description portion of a Gopher menu.

### Syntax

*object*.**ItemDescription**( *index* )

The syntax of the **ItemDescription** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
--------------------	---------------------------

<i>object</i>	A Gopher control.
---------------	-------------------

<i>index</i>	An integer that identifies an element in this property array.
--------------	---

### Remarks

This property holds the human-readable descriptions of a Gopher menu. These descriptions can be used to show the options available to the user.

The *index* ranges from zero (0) to (ItemCount - 1).

This property is only valid after searching using the Action property or GetGopher method.

This property is read-only and only available at run-time.

### Data Type

String

## ItemExtra Property

[See Also](#)

[Error Codes](#)

### Description

Holds any extra data from a Gopher menu item.

### Syntax

*object*.**ItemExtra**( *index* )

The syntax of the **ItemExtra** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
--------------------	---------------------------

<i>object</i>	A Gopher control.
---------------	-------------------

<i>index</i>	An integer that identifies an element in this property array.
--------------	---

### Remarks

This property holds any extra textual data from a Gopher menu. Most menu items have no extra data. There is no defined use for this extra data.

The *index* ranges from zero (0) to (ItemCount - 1).

This property is only valid after searching using the Action property or GetGopher method.

This property is read-only and only available at run-time.

### Data Type

String

## ItemHost Property

[See Also](#)

[Error Codes](#)

### Description

Holds the host name required to load a Gopher item.

### Syntax

*object*.**ItemHost**( *index* )

The syntax of the **ItemHost** property has these parts:

<b>Part</b>	<b>Description</b>
-------------	--------------------

<i>object</i>	A Gopher control.
---------------	-------------------

<i>index</i>	An integer that identifies an element in this property array.
--------------	---

### Remarks

This property contains the name of the host required to select a new Gopher item (for each element in the array). The [ItemHost](#), [ItemPort](#), [ItemType](#) and [ItemSelector](#) properties contain the information required to move to a new Gopher site (by putting the values into the [Host](#), [Port](#), [Type](#) and [Selector](#) properties).

The *index* ranges from zero (0) to ([ItemCount](#) - 1).

This property is only valid after searching using the [Action property](#) or [GetGopher method](#). This property is read-only and only available at run-time.

### Data Type

String

## ItemPort Property

[See Also](#)

[Error Codes](#)

### Description

Holds the host port portion required to move to a new Gopher site/menu.

### Syntax

*object*.**ItemPort**( *index* )

The syntax of the **ItemPort** property has these parts:

<b>Part</b>	<b>Description</b>
<i>object</i>	A Gopher control.
<i>index</i>	An integer that identifies an element in this property array.

### Remarks

This property contains the host port required to select a new Gopher item (for each element in the array). The [ItemHost](#), [ItemPort](#), [ItemType](#) and [ItemSelector](#) properties contain the information required to move to a new Gopher site (by putting the values into the [Host](#), [Port](#), [Type](#) and [Selector](#) properties).

The *index* ranges from zero (0) to ([ItemCount](#) - 1).

This property is only valid after searching using the [Action](#) property or [GetGopher](#) method. This property is read-only and only available at run-time.

### Data Type

Integer



## ItemSelector Property

[See Also](#)

[Error Codes](#)

### Description

Holds the directory name / filename portion of a new site in the loaded Gopher menu.

### Syntax

*object*.**ItemSelector**( *index* )

The syntax of the **ItemSelector** property has these parts:

<b>Part</b>	<b>Description</b>
<i>object</i>	A Gopher control.
<i>index</i>	An integer that identifies an element in this property array.

### Remarks

This property contains the directory or file name on a new host required to select a new Gopher item (for each element in the array). The [ItemHost](#), [ItemPort](#), [ItemType](#) and [ItemSelector](#) properties contain the information required to move to a new Gopher site (by putting the values into the [Host](#), [Port](#), [Type](#) and [Selector](#) properties).

The *index* ranges from zero (0) to ([ItemCount](#) - 1).

This property is only valid after searching using the [Action property](#) or [GetGopher method](#). This property is read-only and only available at run-time.

### Data Type

String

## ItemType Property

[See Also](#)

[Error Codes](#)

### Description

Holds the type of current menu item in the currently loaded Gopher menu.

### Syntax

*object*.**ItemType**( *index* )

The syntax of the **ItemType** property has these parts:

<b>Part</b>	<b>Description</b>
-------------	--------------------

<i>object</i>	A Gopher control.
---------------	-------------------

<i>index</i>	An integer that identifies an element in this property array.
--------------	---

### Remarks

This property contains the type of Gopher item for an entry in the loaded Gopher menu. The [ItemHost](#), [ItemPort](#), [ItemType](#) and [ItemSelector](#) properties contain the information required to move to a new Gopher site (by putting the values into the [Host](#), [Port](#), [Type](#) and [Selector](#) properties).

The *index* ranges from zero (0) to ([ItemCount](#) - 1).

This property is only valid after searching using the [Action property](#) or [GetGopher method](#). This property is read-only and only available at run-time.

### Data Type

String

## LastError Property

[See Also](#)

[Error Codes](#)

### Description

Holds the last error number reported.

### Syntax

*object*.**LastError**

The syntax of the **LastError** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Gopher control.

### Remarks

This property contains the result of the last method or action executed. It is zero if the last method completed without error.

This property is read-only and only available at run-time.

### Data Type

Integer

## OnDownload Event

[See Also](#)

[Error Codes](#)

### Description

OnDownload fires when a file is completely downloaded.

### Syntax

**Sub** *object* **OnDownload**([*index* **As Integer**,] *filename* **As String**)

The syntax of the **OnDownload** event has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Gopher control.
<i>index</i>	An integer that identifies a control if it's in a control array.
<i>filename</i>	A string expression that contains a filename.

### Remarks

This event fires when a file completes downloading. This is either a text file (Type = '0') or any of the binary file types. This event only fires if the download is successful.

*Filename* holds the local filename (see the OutputFilename property) of the file downloaded.

## OutputFilename Property

[See Also](#)

[Error Codes](#)

### Description

Determines the name of the file to use to save retrieved files.

### Syntax

*object*.**OutputFilename** [= *string* ]

The syntax of the **OutputFilename** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Gopher control.
<i>string</i>	A string expression that contains a filename.

### Remarks

When the Gopher control is instructed to download a file from a Gopher server, this is the name of the file (including full path information) where the Gopher control stores the data retrieved. This property is only used for the following Gopher menu Types: 0, 4, 5, 6, 9, g and l.

### Data Type

String

## Port Property

[See Also](#)

[Error Codes](#)

### Description

Determines to which port the Gopher control connects.

### Syntax

*object*.**Port** [= *integer* ]

The syntax of the **Port** property has these parts:

<b>Part</b>	<b>Description</b>
<i>object</i>	A Gopher control.
<i>integer</i>	An integer that contains a port number.

### Remarks

Some gopher sites or items use a different port from the default port 70. This property allows you to set the port to whatever is needed (typically retrieved from the [ItemPort property](#)).

This property defaults to 70.

### Data Type

Integer

## Selector Property

[See Also](#)

[Error Codes](#)

### Description

Specifies the directory or file to retrieve on a [host](#).

### Syntax

*object*.**Selector** [= *string* ]

The syntax of the **Selector** property has these parts:

<b>Part</b>	<b>Description</b>
<i>object</i>	A Gopher control.
<i>string</i>	A string expression that contains a string which specifies a directory.

### Remarks

This property determines which directory or file to retrieve from a Gopher site. This property, combined with [Host](#), [Port](#) and [Type](#), determines exactly what the Gopher control retrieves through the [GetGopher method](#) or [Action property](#).

If this property is set to an empty string, the [GetGopher method](#) (or its [Action property](#) equivalent) will retrieve the main Gopher menu for the [Host](#) specified.

### Data Type

String

## Type Property

[See Also](#)

[Error Codes](#)

### Description

Specifies the type of directory or file to retrieve on a [host](#).

### Syntax

*object.Type* [= *string* ]

The syntax of the **Type** property has these parts:

<b>Part</b>	<b>Description</b>
<i>object</i>	A Gopher control.
<i>string</i>	A string expression that determines the type of directory or file to retrieve.

### Remarks

This property determines what type of directory or file to retrieve from a Gopher site. This property, combined with [Host](#), [Port](#) and [Selector](#), determines exactly what the Gopher control retrieves through the [GetGopher method](#) or [Action property](#).

This property must be one of the following values:

<b>Value</b>	<b>Description</b>
0	The item is a text file entity. Gopher will retrieve the text file and store it in the file specified by the <a href="#">OutputFilename property</a> .
1	The item is a Gopher menu. Gopher will retrieve the menu into the ItemXXX properties.
2	The information is a CSO phone book entity which is accessed through the CSO protocol. The Gopher Control does not support this.
3	This is an error.
4	This is a Macintosh file encoded in the BinHex format.
5	This is a binary file. Gopher will retrieve the text file and store it in the file specified by the <a href="#">OutputFilename property</a> .
6	This type is a UUencoded file.
7	This is a full text server. The <a href="#">Selector property</a> should contain the word(s) to search for. Gopher will retrieve all of the found items in the ItemXXX properties as a normal Gopher menu.
8	This would open a Telnet session. The Gopher control does not support this type.
9	This is a binary file. Gopher will retrieve the text file and store it in the file specified by the <a href="#">OutputFilename property</a> .
g	This item is a GIF file. Gopher will retrieve the text file and store it in the file specified by the <a href="#">OutputFilename property</a> .
l	This type specifies an image file (format unknown). Gopher will retrieve the text file and store it in the file specified by the <a href="#">OutputFilename property</a> .

### Data Type

String



## Version Property

Error Codes

### Description

Shows the version of the control.

### Syntax

*object*.**Version**

The syntax of the **Version** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Gopher control.

### Remarks

This property holds the current version of the control. It is read-only and available at both design-time and run-time.

### Data Type

String

## **Getting Custom Controls Written**

If you or your organization would like to have custom controls written, you can contact us at the following:

Mabry Software, Inc.  
Post Office Box 31926  
Seattle, WA 98103-1926  
Phone: 206-634-1443  
Fax: 206-632-0272 or 206-364-3196  
Internet: [mabry@mabry.com](mailto:mabry@mabry.com)

You can also contact Zane Thomas. He can be reached at:

Zane Thomas  
Post Office Box 121  
Indianola, WA 98342  
Internet: [zane@mabry.com](mailto:zane@mabry.com)

## **Licensing Information**

### **Legalese Version**

Mabry Software grants a license to use the enclosed software to the original purchaser. Copies may be made for back-up purposes only. Copies made for any other purpose are expressly prohibited, and adherence to this requirement is the sole responsibility of the purchaser.

Customer written executable applications containing embedded Mabry products may be freely distributed, without royalty payments to Mabry Software, provided that such distributed Mabry product is bound into these applications in such a way so as to prohibit separate use in design mode, and that such Mabry product is distributed only in conjunction with the customers own software product. The Mabry Software product may not be distributed by itself in any form.

Neither source code for Mabry Software products nor modified source code for Mabry Software products may be distributed under any circumstances, nor may you distribute .OBJ, .LIB, etc. files that contain our routines. This control may be used as a constituent control only if the compound control thus created is distributed with and as an integral part of an application. Permission to use this control as a constituent control does not grant a right to distribute the license (LIC) file or any other file other than the control executable itself. This license may be transferred to a third party only if all existing copies of the software and its documentation are also transferred.

This product is licensed for use by only one developer at a time. Mabry Software expressly prohibits installing this product on more than one computer if there is any chance that both copies will be used simultaneously. This restriction also extends to installation on a network server, if more than one workstation will be accessing the product. All developers working on a project which includes a Mabry Software product, even though not working directly with the Mabry product, are required to purchase a license for that Mabry product.

This software is provided as is. Mabry Software makes no warranty, expressed or implied, with regard to the software. All implied warranties, including the warranties of merchantability and fitness for a particular use, are hereby excluded.

MABRY SOFTWARE'S LIABILITY IS LIMITED TO THE PURCHASE PRICE. Under no circumstances shall Mabry Software or the authors of this product be liable for any incidental or consequential damages, nor for any damages in excess of the original purchase price.

To be eligible for free technical support by telephone, the Internet, CompuServe, etc. and to ensure that you are notified of any future updates, please complete the enclosed registration card and return it to Mabry Software.

### **English Version**

We require that you purchase one copy of a control per developer on a project. If this is met, you may distribute the control with your application royalty free. You may never distribute the LIC file. You may not change the product in any way that removes or changes the requirement of a license file.

We encourage the use of our controls as constituent controls when the compound controls you create are an integral part of your application. But we don't allow distribution of our controls as constituents of other controls when the compound control is not part of an application. The reason we need to have this restriction is that without it someone might decide to use our control as a constituent, add some trivial (or even non-trivial) enhancements and then sell the compound control. Obviously there would be little difference between that and just plain reselling our control.

If you have purchased the source code, you may not re-distribute the source code either (nor may you copy it into your own project). Mabry Software retains the copyright to the

source code.

Your license is transferable. The original purchaser of the product must make the transfer request. Contact us for further information.

The sample versions of our products are intended for evaluation purposes only. You may not use the sample version to develop completed applications.

