Help for Time

Registration Information

Order Form

Getting Custom Controls Written

Licensing Information

Description

The Time control allows you to retrieve the current Greenwich Mean Time (GMT) from time servers on the Internet.   The time servers usually have the correct time, since they all talk among themselves to keep things accurate.   Our favorite server is the one at MIT (mit.edu).   Another good server is at "tock.usno.navy.mil".

You can use the Time control in any application where you need to have accurate time/date stamps.   This includes financial transactions, medical data, long-term data collection, or anywhere else that a reliable time-stamp is needed.   You can even use this control to update the time-of-day on your systems and servers.

The Time control requests the time from time servers that use the Network Time Protocol (RFC 1305) to synchronize themselves.

File Name

TIME1.VBX, TIME32.OCX

ActiveX / OCX Object Name

Mabry.TimeCtrl

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

MabryTime

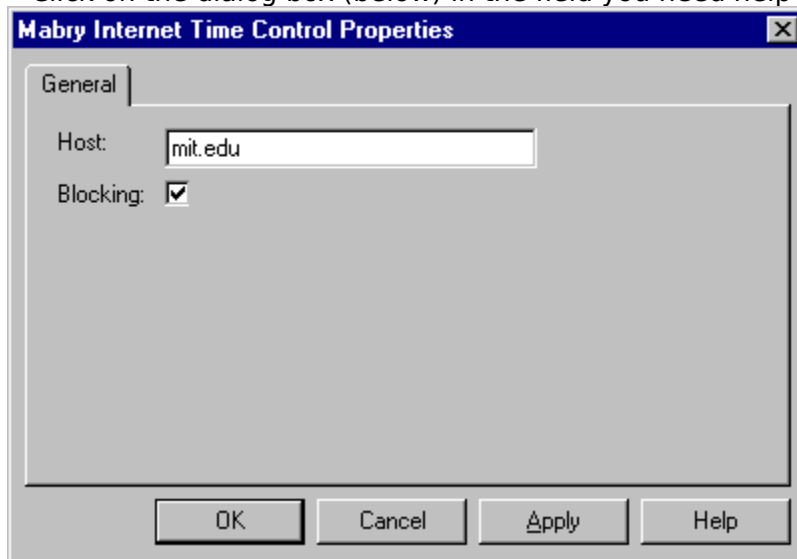## VBX Compatibility

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.

Time Property Page

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

Mabry Internet Time Control Properties ☒

General

Host: | mit.edu

Blocking: ☑

[OK] [Cancel] [Apply] [Help]

Time 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
*__GMTTime__ Property
*__Host__ Property
*__LastError__ Property
*__LocalTime__ Property
*__Version__ Property

Time 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

Time Methods

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

*__AboutBox__ Method
*__GetTime__ Method

Frequently Asked Questions

General Questions

How do I set the system time using the Mabry Time control?

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?

How do I set the system time using the Mabry Time control?

Use the LocalTime property to set Visual Basic's Time statement.   For example:

```
Time = Time1.LocalTime
```

If you want to do a bit of validation first:

```
If IsDate(Time1.LocalTime) Then
    Time = Time1.LocalTime
End If
```

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

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?

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?

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?

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?

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)?

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?

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_octect 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
    ' 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"?

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?

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?

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 likelyhood 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?

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"?

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 likelyhood 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?

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?

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

Time was written by Zane Thomas.

CONTACT INFORMATION

Orders, inquiries, technical support, questions, comments, etc. can be sent to 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 Time (control only) is US$20 (US$25 for International orders).   The cost of Time and the C/C++ source code (of the control itself) is US$50 (US$55 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.   Time's SWREG ID number is 6524.   The source code version's ID number is 9063.   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.

Time Order Form

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

Mail this Mabry Software, Inc.
form to: 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 Time?

_____

Name: _____

Ship to: _____

_____

_____

_____

Phone: _____

Fax: _____

E-Mail: _____

MC/VISA/AMEX: _____ exp. _____

P.O. # (if any): _____ Signature _____

qty ordered ____   REGISTRATION
$20.00 ($25.00 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
$50.00 ($55.00 international).   Check or money order in U.S.
currency drawn on a U.S. bank.   Add $5.00 per order for shipping
and handling.

See Also

  AsyncError Event

  Done Event

  GetTime Method

See Also

Action Property

Done Event

GetTime Method

See Also

Action Property

AsyncError Event

Done Event

GetTime Method

See Also

  Action Property
  Blocking Property
  GMTTime Property
  LocalTime Property
  AsyncError Event
  GetTime Method

See Also

Action Property

Blocking Property

Host Property

GMTTime Property

LocalTime Property

Done Event

See Also

[**Action** Property](#)

[**LocalTime** Property](#)

[**GetTime** Method](#)

See Also

[**Action** Property](#)
[**GetTime** Method](#)

See Also

Action Property

AsyncError Event

Done Event

GetTime Method

See Also

Action Property

GMTTime Property

GetTime Method

AboutBox Method

Description

displays the About Box for the control.

Syntax

*object.***AboutBox**

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

Part	Description
object	Required. A Time control.

Remarks

This method displays the About Box for this control to display copyright information.

Action Property

Description

Causes control to initiate a command / method.

Syntax

*object.***Action** [= *action*]

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

Part	Description
object	A Time control.
action	An integer that determines the action to take.

Remarks

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

Value	Description
1	Retrieves the time. This works just like the GetTime method.

Data Type

Integer (enumerated)

AsyncError Event

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:

Part	Description
object	A Time control.
index	An integer that identifies a control if it's in a control array.
errornumber	An integer that holds the current error number.
errormessage	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

Description

Determines if any methods or actions are blocking.

Syntax

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

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

Part	Description
object	A Time control.
boolean	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 any of the methods 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.   When the user-interface requires that the user be able to cancel file transfers, it is recommended that Blocking be set to False so that your program can respond quickly to user actions.

Data Type

Integer (boolean)

Done Event

Description

This event procedure is fired when the control retrieves the current time.

Syntax

Sub *object*_**Done(**[*index* **As Integer**,] *errornumber* **As Integer)**

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

Part	Description
object	A Time control.
index	An integer that identifies a control if it's in a control array.
errornumber	An integer that is always zero. This can be ignored.

Remarks

This event fires when an asynchronous time retrieval operation completes (when <u>Blocking</u> is set to False).   The current time will be found in the <u>GMTTime</u> and <u>LocalTime</u> properties. They are formatted as follows:

```
mon-dd-yyyy hh:mm:ss
```

For example:

```
Feb-26-1995 08:25:22
Aug-14-1996 10:30:58
```

GetTime Method

Description

Gets current time from specified server.

Syntax

*object.***GetTime** *host*, *blocking*

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

Part	Description
object	Required. A Time control.
host	Optional. A string that holds either the name of the time host or the IP address of the time host.
blocking	Optional. A boolean flag that determines if the method should return immediately (False) or if the control should wait until completion to return (True).

Remarks

This method requests the current time from the host.   If no host is specified, the Host property is used.

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 Time control does not return control to your program until it retrieves the time 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, a warning   is reported (10035) immediately.   This is only a warning from the WinSock layer letting you know that transmission to the host needs to wait, briefly, and can be treated as normal.

GMTTime Property

Description

Holds the last time retrieved from a time server (Greenwich Mean Time).

Syntax

*object.***GMTTime**

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

Part	Description
object	A Time control.

Remarks

This property is valid only after a <u>GetTime method</u> or <u>Action property</u> command retrieves the time from a time server.   GMTTime is formatted as follows :

```
mon-dd-yyyy hh:mm:ss
```

For example:

```
Feb-26-1995 08:25:22
Aug-14-1996 10:30:58
```

GMTTime's format is compatible with Visual Basic's CDate, CVDate, IsTime and related functions.   This property is only available at run-time and is read-only.

Data Type

String

Host Property

Description

Time requests information from the host with this address.

Syntax

*object.***Host** [= *hostname*]

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

Part	Description
object	A Time control.
hostname	A string expression that holds the name or IP address of a time server.

Remarks

This property determines which server that the Time control polls for the Greenwich Mean Time.   The time is retrieved when you use either the <u>Action property</u> or the <u>GetTime method</u>.

Data Type

String

LastError Property

Description

Holds the last error number reported.

Syntax

*object.***LastError**

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

Part	**Description**
object	A Time control.

Remarks

This property contains the result of the last method executed.   It is zero if the last method completed without error.   Otherwise, it contains an error code.

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

Data Type

Integer

LocalTime Property

Description

Holds the last time retrieved from a time server (based on your local time zone).

Syntax

*object.***LocalTime**

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

Part	Description
object	A Time control.

Remarks

This property is valid only after a <u>GetTime method</u> or <u>Action property</u> command retrieves the time from a time server.   LocalTime is formatted as follows :

```
mon-dd-yyyy hh:mm:ss
```

For example:

```
Feb-26-1995 08:25:22
Aug-14-1996 10:30:58
```

LocalTime's format is compatible with Visual Basic's CDate, CVDate, IsTime and related functions.   This property is only available at run-time and is read-only.

Data Type

String

Version Property

Description

Shows the version of the control.

Syntax

*object.***Version**

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

Part	**Description**
object	A Time 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

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

Zane Thomas
Post Office Box 121
Indianola, WA   98342

Internet: 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.