Help for Slider Controls

Properties Events

Registration Information

Order Form

Getting Custom Controls Written

Licensing Information

Description

This are sliders control. They are similar in function to scroll bars, but have a different appearance.

Easily create a cool looking audio mixer with the Slider horizontal and vertical custom controls. A complete set of bevel properties allows you to quickly get that 3D look. Four different thumb styles adds to the flexibility of this control. Tickmark properties round out this controls great features.

- Complete set of bevel properties
- Slider control is similar in function to a scroll bar, but looks better
- Pro audio mixer fader style thumbs
- Up, down, left and right thumb properties
- Left, right and both tick properties with tickcolor, tickcount, tickwidth, ticklength and gap control
- Control of the 3-D style of the track

VBX Compatibility

VB 2.0, 3.0 and 4.0 (16-bit)

File Name

HSLIDE.VBX, HSLIDE16.OCX, HSLIDE32.OCX
VSLIDE.VBX, VSLIDE16.OCX, VSLIDE32.OCX

Object Type

HSlider
VSlider

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

Registration Information

Credits

The MIDI Pack was written by James Shields and Zane Thomas. The slider controls were written by James Shields.

Inquiries, tech support, comments should be sent to Mabry Software. Our address is 71231,2066 on CompuServe, or mabry@mabry.com on Internet. You can call us at 2066341443 or fax us at 206632-0272. If you need to send something via U.S. Mail, the address is:

Mabry Software, Inc.
Post Office Box 31926
Seattle, WA 98103-1926

Registration

You can register this program by sending $20 ($25 for international orders) and your address. You can register the Slider controls **and** their C++ source code by sending $45 ($50 for international orders). With your order, you will receive a copy of our manual documenting all of the MIDI Pack controls.

Add $5 per order for shipping and handling.

For your convenience, an order form has been provided that you can print out directly from this help file.

Prices subject to change without notice.

E-mail Discount

You may take a $5 discount for e-mail delivery of this package (CompuServe or Internet). If you choose this option, please note: a printed manual is not included. 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 members may also take the $5 e-mail discount by registering this package in the software registration forum (GO SWREG). Sliders SWREG ID number is 10286. The source code version's ID number is 10288. 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.

Credit Card Orders

We accept VISA, Mastercard and American Express. If you e-mail your order to us, please be sure to include your card number, expiration date, complete mailing address, and your phone number (in case we have any questions about your order).

Slider 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
CompuServe: 71231,2066
Internet: mabry@mabry.com
Web: www.mabry.com

Where did you get this copy of Slider?

_____

Ship to: _____

_____

_____

_____

_____

Phone: _____

Fax: _____

E-Mail: _____

MC/VISA/AMEX: _____ exp. _____

P.O. # (if any): _____

qty ordered ____ REGISTRATION
$20 each ($25 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
$45 each ($50 international). Check or money order in U.S. currency drawn on a U.S. bank. Add $5.00 per order for shipping and handling.

Slider Properties

All of the properties that apply to this control are in this table. Properties that have special meaning for this control or that only apply to this control are marked with an asterisk (*).

BackColor Property
***BevelInner** Property
***BevelOuter** Property
***BevelWidth** Property
***BorderWidth** Property
Enabled Property
***Gap** Property
Height Property
hWnd Property
Index Property
***LargeChange** Property
Left Property
***LinkControl** Property
***LinkProperty** Property
***Max** Property
***Min** Property
Name Property
Parent Property
Tag Property
***ThumbHeight** Property
***ThumbStyle** Property
***ThumbWidth** Property
***TickColor** Property
***TickCount** Property
***TickLength** Property
***TickMarks** Property
***TickWidth** Property
Top Property
***TrackBevel** Property
***TrackWidth** Property
***Value** Property (default)
Visible Property
Width Property

Slider Events

All of the events that apply to this control are in this table. Events that have special meaning for this control or that only apply to this control are marked with an asterisk (*).

***Change** Event

GotFocus Event

LostFocus Event

MouseDown Event

MouseMove Event

MouseUp Event

***Scroll** Event

BevelInner Property

See Also Example

Description

Determines the 3-D style of the border immediately surrounding the control.

Usage

[*form.*][*control.*]**BevelInner**[= *integer*]

Remarks

The value of this property determines the style of the inner border. This property may be one of four values:

Constant	Value	Description
bcNone	0	Normal frame
bcRaised	1	Raised frame (3-D)
bcInset	2	Inset frame (3-D)
bcLowered	3	Lowered frame (3-D)

Data Type

Integer (enumerated)

See Also

Properties:

- BevelOuter
- BevelWidth
- BorderWidth

BevelOuter Property

See Also Example

Description

Determines the 3-D style of the border (if any) surrounding the control.

Usage

[*form.*][*control.*]**BevelOuter**[= *integer*]

Remarks

The value of this property determines the style of the control's border. This property may be one of four values:

Constant	Value	Description
bcNone	0	Normal frame
bcRaised	1	Raised frame (3-D)
bcInset	2	Inset frame (3-D)
bcLowered	3	Lowered frame (3-D)

Data Type

Integer (enumerated)

See Also

Properties:

- BevelInner
- BevelWidth
- BorderWidth

BevelWidth Property

See Also Example

Description

Determines the width of the inner and outer borders (bevels).

Usage

[*form.*][*control.*]**BevelWidth**[= *integer*]

Remarks

The value of this property determines the width of the inner border (if any, see BevelInner) and the outer border (if any, BevelOuter). This is always measured in pixels.

Data Type

Integer

See Also

Properties:

- BevelInner
- BevelOuter
- BorderWidth

BorderWidth Property

See Also Example

Description

Determines the distance between the inner border and the outer border.

Usage

[*form.*][*control.*]**BorderWidth**[= *integer*]

Remarks

The value of this property determines the distance between the outer border (if any, see BevelOuter) and the inner border (if any, see BevelInner). This is always measured in pixels.

Data Type

Integer

See Also

Properties:

BevelInner

BevelOuter

Change Event

See Also

Description

Occurs when the value has changed.

Syntax

Sub *ctlname*_**Change ( )**

Remarks

This event occurs when the value of the control has changed (usually through user interaction). When this event occurs, the control also updates the control specified by the link properties.

See Also

Events:

- Scroll

Properties:

- LinkControl
- LinkProperty
- Value

Gap Property

See Also Example

Description

Determines the distance between the inside of the border and the tick marks.

Usage

[*form.*][*control.*]**Gap**[= *integer*]

Remarks

The value of this property determines the distance between the inner border and the tick marks. This property is measured in pixels.

Data Type

Integer

See Also

Properties:

- BevelInner
- BevelOuter
- BevelWidth
- BorderWidth

LargeChange Property

Description

Determines the how far the slider moves when clicked outside the thumb..

Usage

[*form.*][*control.*]**LargeChange**[= *integer*]

Remarks

The value of this property determines how far the thumb moves when the control is clicked outside the thumb and near the track.

Data Type

Integer

LinkControl and LinkProperty Properties

Description

Sets up link to another control.

Usage

[*form.*][*control.*]**LinkControl**
[*form.*][*control.*]**LinkProperty**

Remarks

These properties set up a link with another control. When the Value property changes, the control sends the new value to the control and property specified by these properties. If the other control is one of the controls in this package (i.e., Horizontal Indicator, Horizontal Slider, Knob, MIDI File, MIDI Input, MIDI Output, Vertical Indicator, or Vertical Slider), the current control's Value property will be updated when the other control's property changes.

At design-time, be sure to set the LinkControl property first. The LinkProperty combo box will display all of the valid properties for that control.

These properties are changable at design-time, and read-only at run-time.

IMPORTANT NOTE: These properties are only present in the VBX versions of these controls.

Data Type

String

Min and Max Properties

See Also

Description

Determines the range of values for this control.

Usage

[*form.*][*control.*]**Max**[= *integer*]
[*form.*][*control.*]**Min**[= *integer*]

Remarks

These properties determine the range of values for the control in question. If Max is set to less than Min, then the range of values is swapped.

Data Type

Integer

See Also

Properties:

Value

Scroll Event

See Also

Description

Occurs while a user changes the value.

Syntax

Sub *ctlname*_**Scroll ( )**

Remarks

You can use this event to perform calculations or to manipulate controls that must be coordinated with changes in these controls. Use the Change event when you want an update to occur after the change is complete.

See Also

Events:

Change

Properties:

Value

ThumbHeight and ThumbWidth Properties

See Also Example

Description

Determines the size of the thumb.

Usage

[*form.*][*control.*]**ThumbHeight**[= *height*]
[*form.*][*control.*]**ThumbWidth**[= *width*]

Remarks

The value of these properties determine the size of the thumb. These properties are measured in twips.

Data Type

Real

See Also

Properties:

ThumbStyle

ThumbStyle Property

See Also Example

Description

Determines the style of the thumb.

Usage

[*form.*][*control.*]**ThumbStyle**[= *integer*]

Remarks

The value of this property determines the style of the control's border. This property may be one of four values:

Constant	Value	Description
tscNormal	0	Normal
tscPointedUp / **tscPointedLeft**	1	Pointer up/left
tscPointedDown / **tscPointedRight**	2	Pointed down/right
tscLined	3	Lined

Data Type

Integer (enumerated)

See Also

Properties:

ThumbHeight

ThumbWidth

TickColor Property

See Also Example

Description

Determines what color the ticks will be.

Usage

[*form.*][*control.*]**TickColor**[= *color*]

Remarks

This property specifies the color of the tick marks.

Data Type

Color

See Also

Properties:

- TickCount
- TickLength
- TickWidth

TickCount Property

See Also Example

Description

Determines how many tick marks there will be.

Usage

[*form.*][*control.*]**TickCount**[= *integer*]

Remarks

This property determines how many tick marks there will be.

Data Type

Integer

See Also

Properties:

- TickColor
- TickLength
- TickWidth

TickLength Property

See Also Example

Description

Determines the length of the tick marks.

Usage

[*form.*][*control.*]**TickLength**[= *integer*]

Remarks

This property specifies the length, in pixels, of the tick marks.

Data Type

Integer

See Also

Properties:

- TickColor
- TickCount
- TickWidth

TickMarks Property

See Also

Description

Determines where the ticks will appear.

Usage

[*form.*][*control.*]**TickMarks**[= *integer*]

Remarks

This property where the tick marks will be. The legitimate values are:

Constant	Value	Meaning
vtmcNone / **htmcNone**	0	No tick marks
vtmcLeft / **htmcTop**	1	Top for HSlider, Left for VSlider
vtmcRight / **htmcBottom**	2	Bottom for HSlider, Right for VSlider
vtmcBoth / **htmcBoth**	3	Both

Data Type

Integer

See Also

Properties:

- TickColor
- TickCount
- TickLength
- TickWidth

TickWidth Property

See Also Example

Description

Determines the width of the tick marks.

Usage

[*form.*][*control.*]**TickWidth**[= *integer*]

Remarks

This property determines the width of the tick marks. This property is measured in pixels.

Data Type

Integer

See Also

Properties:

- TickColor
- TickCount
- TickLength

TrackBevel Property

See Also Example

Description

Determines the 3-D style of the track.

Usage

[*form.*][*control.*]**TrackBevel**[= *integer*]

Remarks

The value of this property determines the style of the control's border. This property may be one of four values:

Constant	Value	Description
bcNone	0	Normal
bcRaised	1	Raised (3-D)
bcInset	2	Inset (3-D)
bcLowered	3	Lowered (3-D)

Data Type

Integer (enumerated)

See Also

Properties:

TrackWidth

TrackWidth Property

See Also Example

Description

Determines the width of the track.

Usage

[*form.*][*control.*]**TrackWidth**[= *integer*]

Remarks

The value of this property determines the width of the track. This property is measured in pixels.

Data Type

Integer

See Also

Properties:

TrackBevel

Value Property

See Also

Description

Specifies the current position of the control.

Usage

[*form.*][*control.*]**Value**[= *integer*]

Remarks

This property determines the current value of the control. This is the default property of these controls.

Data Type

Integer

See Also

Events:

- Change
- Scroll

Properties:

- LinkControl
- LinkProperty
- Max
- Min

Close Copy Print

Bevel Properties Example

In this example, the program shows what happens when you vary the bevels on the controls. To try this example, paste the code into the Declarations section of a form that contains a knob, a horizontal indicator, and a horizontal slider control. Press F5. Play with the knob.

```
Sub Form_Load ()
  Form1.BackColor = &HC0C0C0

  Knob1.Width = 3000
  Knob1.Height = 2000
  Knob1.Radius = 500
  Knob1.TickCount = 4
  Knob1.Min = 0
  Knob1.Max = 3
  Knob1.Value = 0
  Knob1.FontSize = 7
  Knob1.FontBold = False
  Knob1.FontName = "Arial"
  Knob1.FontSize = 7
  Knob1.TickCaption(0) = "None"
  Knob1.TickCaption(1) = "Raised"
  Knob1.TickCaption(2) = "Inset"
  Knob1.TickCaption(3) = "Lowered"

  HIndicator1.BackColor = &HC0C0C0

  HSlider1.TrackBevel = 0
  HSlider1.TrackWidth = 5
  HSlider1.BorderWidth = 4
End Sub

Sub Knob1_Scroll ()
  HSlider1.BevelInner = Knob1.Value
  HSlider1.BevelOuter = Knob1.Value
  HIndicator1.BevelInner = Knob1.Value
  HIndicator1.BevelOuter = Knob1.Value
End Sub
```

Close | Copy | Print

Gap Property Example

In this example, the program shows what happens when you vary the gap. To try this example, paste the code into the Declarations section of a form that contains a horizontal scroll bar, a label, and a horizontal slider control. Press F5. Play with the horizontal scroll bar.

```
Sub Form_Load ()
  Form1.BackColor = &HC0C0C0

  Label1.BackColor = &HC0C0C0
  Label1.Top = 240
  Label1.Left = 2840
  Label1.Height = 255

  HSlider1.Height = 1000
  HSlider1.Width = 2000

  HScroll1.Top = 240
  HScroll1.Left = 720
  HScroll1.Width = 2000
  HScroll1.Min = 0
  HScroll1.Max = 20
  HScroll1.Value = 2

  HSlider1.BevelOuter = 1
  HSlider1.BevelInner = 3
  HSlider1.TickMarks = 3
  HSlider1.TickCount = 11
  HSlider1.Height = 1000
  HSlider1.Width = 2000
  HSlider1.ThumbHeight = 360
  HSlider1.ThumbWidth = 120
  HSlider1.Gap = HScroll1.Value
  HSlider1.Value = 50
End Sub

Sub HScroll1_Change ()
  Call HScroll1_Scroll
End Sub

Sub HScroll1_Scroll ()
  HSlider1.Gap = HScroll1.Value
  Label1.Caption = "Gap: " & HScroll1.Value
End Sub
```

Close Copy Print

ThumbHeight, ThumbStyle, and ThumbWidth Properties Example

In this example, the program shows what happens when you vary the size of the thumb. To try this example, paste the code into the Declarations section of a form that contains a horizontal slider, a horizontal scroll bar, a vertical scroll bar, a knob, and two label controls. Press F5. Play with the scroll bars and the knob.

```
Sub Form_Load ()
 Form1.BackColor = &HC0C0C0
 Form1.Height = 4880
 Form1.Width = 4000

 Knob1.Left = 204
 Knob1.Top = 2400
 Knob1.Width = 3400
 Knob1.Height = 2000
 Knob1.Radius = 500
 Knob1.Min = 0
 Knob1.Max = 3
 Knob1.TickCount = 4
 Knob1.TickCaption(0) = "Normal"
 Knob1.TickCaption(1) = "Pointed Up"
 Knob1.TickCaption(2) = "Pointed Down"
 Knob1.TickCaption(3) = "Lined"

 Label1.BackColor = &HC0C0C0
 Label1.Top = 240
 Label1.Left = 2840
 Label1.Height = 255

 Label2.BackColor = &HC0C0C0
 Label2.Top = 1840
 Label2.Left = 240
 Label2.Height = 255

 HSlider1.Height = 1000
 HSlider1.Width = 2000

 HScroll1.Top = 240
 HScroll1.Left = 720
 HScroll1.Width = 2000
 HScroll1.Min = 90
 HScroll1.Max = 500
 HScroll1.Value = 120

 VScroll1.Top = 720
 VScroll1.Left = 240
 VScroll1.Height = 1000
 VScroll1.Min = 90
 VScroll1.Max = 500
 VScroll1.Value = 240

 HSlider1.Height = 1000
 HSlider1.Width = 2000
```

```
 HSlider1.ThumbHeight = VScroll1.Value
 HSlider1.ThumbWidth = HScroll1.Value
 HSlider1.Value = 50
End Sub

Sub HScroll1_Change ()
 Call HScroll1_Scroll
End Sub

Sub HScroll1_Scroll ()
 HSlider1.ThumbWidth = HScroll1.Value
 Label1.Caption = HScroll1.Value
End Sub

Sub Knob1_Change ()
 Call Knob1_Scroll
End Sub

Sub Knob1_Scroll ()
 HSlider1.ThumbStyle = Knob1.Value
End Sub

Sub VScroll1_Change ()
 Call VScroll1_Scroll
End Sub

Sub VScroll1_Scroll ()
 HSlider1.ThumbHeight = VScroll1.Value
 Label2.Caption = VScroll1.Value
End Sub
```

Close | Copy | Print

Tick Properties Example

In this example, the program shows what happens when you change the look of the tick marks. To try this example, paste the code into the Declarations section of a form that contains a horizontal slider, a knob, two command buttons, four horizontal scroll bars, four labels, and a common dialog control. Press F5. Play with the scroll bars and the command buttons.

```
Sub Command1_Click ()
  CMDialog1.Color = HSlider1.TickColor
  CMDialog1.Flags = 1
  CMDialog1.Action = 3

  HSlider1.TickColor = CMDialog1.Color
  Knob1.TickColor = CMDialog1.Color
End Sub

Sub Command2_Click ()
  CMDialog1.Color = Knob1.TickCaptionColor
  CMDialog1.Flags = 1
  CMDialog1.Action = 3

  Knob1.TickCaptionColor = CMDialog1.Color
End Sub

Sub Form_Load ()
  Form1.BackColor = &HC0C0C0

  HSlider1.Top = 1680
  HSlider1.Left = 240
  HSlider1.Width = 6000
  HSlider1.Height = 600
  HSlider1.Value = 100
  HSlider1.BackColor = &HC0C0C0

  Knob1.Top = 2400
  Knob1.Left = 240
  Knob1.Width = 1800
  Knob1.Height = 1800
  Knob1.Radius = 400
  Knob1.TickCount = 5

  Command1.Top = 240
  Command1.Left = 240
  Command1.Width = 1800
  Command1.Height = 360
  Command1.Caption = "Change TickColor"

  Command2.Top = 720
  Command2.Left = 240
  Command2.Width = 1800
  Command2.Height = 360
  Command2.Caption = "Change TickCaptionColor"

  HScroll1.Top = 240
```

```
  HScroll1.Left = 2160
  HScroll1.Width = 900
  HScroll1.Min = 0
  HScroll1.Max = 20
  HScroll1.Value = Knob1.TickCount

  Label1.Top = 240
  Label1.Left = 3180
  Label1.Width = 2000
  Label1.BackColor = &HC0C0C0

  HScroll2.Top = 600
  HScroll2.Left = 2160
  HScroll2.Width = 900
  HScroll2.Min = 0
  HScroll2.Max = 20
  HScroll2.Value = Knob1.TickGap

  Label2.Top = 600
  Label2.Left = 3180
  Label2.Width = 2000
  Label2.BackColor = &HC0C0C0

  HScroll3.Top = 960
  HScroll3.Left = 2160
  HScroll3.Width = 900
  HScroll3.Min = 0
  HScroll3.Max = 20
  HScroll3.Value = Knob1.TickLength

  Label3.Top = 960
  Label3.Left = 3180
  Label3.Width = 2000
  Label3.BackColor = &HC0C0C0

  HScroll4.Top = 1320
  HScroll4.Left = 2160
  HScroll4.Width = 900
  HScroll4.Min = 0
  HScroll4.Max = 20
  HScroll4.Value = Knob1.TickWidth

  Label4.Top = 1320
  Label4.Left = 3180
  Label4.Width = 2000
  Label4.BackColor = &HC0C0C0
End Sub

Sub HScroll1_Change ()
  Dim I As Integer

  HSlider1.TickCount = HScroll1.Value
  Knob1.TickCount = HScroll1.Value
  Label1.Caption = "TickCount: " & HScroll1.Value

  For I = 0 To HScroll1.Value - 1
     Knob1.TickCaption(I) = Chr$(I + 65)
```

```
 Next I
End Sub

Sub HScroll1_Scroll ()
 Call HScroll1_Change
End Sub

Sub HScroll2_Change ()
 HSlider1.Gap = HScroll2.Value
 Knob1.TickGap = HScroll2.Value
 Label2.Caption = "TickGap: " & HScroll2.Value
End Sub

Sub HScroll2_Scroll ()
 Call HScroll2_Change
End Sub

Sub HScroll3_Change ()
 HSlider1.TickLength = HScroll3.Value
 Knob1.TickLength = HScroll3.Value
 Label3.Caption = "TickLength: " & HScroll3.Value
End Sub

Sub HScroll3_Scroll ()
 Call HScroll3_Change
End Sub

Sub HScroll4_Change ()
 HSlider1.TickWidth = HScroll4.Value
 Knob1.TickWidth = HScroll4.Value
 Label4.Caption = "TickWidth: " & HScroll4.Value
End Sub

Sub HScroll4_Scroll ()
 Call HScroll4_Change
End Sub
```

Close Copy Print

TrackBevel Property Example

In this example, the program shows what happens when you vary the track bevel. To try this example, paste the code into the Declarations section of a form that contains a knob, and a horizontal slider control. Press F5. Play with the knob.

```
Sub Form_Load ()
  Form1.BackColor = &HC0C0C0

  Knob1.Width = 3000
  Knob1.Height = 2000
  Knob1.Radius = 500
  Knob1.TickCount = 4
  Knob1.Min = 0
  Knob1.Max = 3
  Knob1.Value = 0
  Knob1.FontSize = 7
  Knob1.FontBold = False
  Knob1.FontName = "Arial"
  Knob1.FontSize = 7
  Knob1.TickCaption(0) = "Normal"
  Knob1.TickCaption(1) = "Raised"
  Knob1.TickCaption(2) = "Inset"
  Knob1.TickCaption(3) = "Lowered"

  HSlider1.TrackBevel = 0
  HSlider1.TrackWidth = 5
End Sub

Sub Knob1_Scroll ()
  HSlider1.TrackBevel = Knob1.Value
End Sub
```

Close Copy Print

TrackWidth Property Example

In this example, the program shows what happens when you vary the track width. To try this example, paste the code into the Declarations section of a form that contains a label, a vertical scroll bar, and a horizontal slider control. Press F5. Play with the scroll bar.

```
Sub Form_Load ()
 Label1.Caption = "0"

 HSlider1.TrackBevel = 3

 VScroll1.Min = 0
 VScroll1.Max = 20
End Sub

Sub VScroll1_Scroll ()
 Label1.Caption = VScroll1.Value
 HSlider1.TrackWidth = VScroll1.Value
End Sub
```

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

CompuServe: 71231,2066
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.