Help for LED

Registration Information

Order Form

Getting Custom Controls Written

Description

LED is a Visual Basic custom control that behaves like an LED.   3-D effects and colors are all user definable.

File Name

LED1.VBX

Object Type

MabryLED

Compatibility

VB 1.0 and up

Distribution Note   When you develop and distribute an application that uses LED, you should install the file LED1.VBX into the users Windows SYSTEM directory.   LED has version information built into it.   So, during installation, you should ensure that you are not overwriting a newer version of LED.

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	*DisabledColor	*OnColor
*BevelInner	Enabled	Parent
*BevelInnerShape		Height	Tag
*BevelOuter	hWnd	Top
*BevelOuterShape		Index	*Value
*BevelWidth	Left	Visible
*Border	Name	Width
*BorderWidth	*OffColor	

Value is the default value for the control.

BevelInner Property

Description

Determines the 3-D style of the border immediately surrounding the LED area.

Usage

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

Remarks

The value of this property determines the style of the LED's border.   If the Border property is set to False, this property is ignored.   This property may be one of four values:

Value	Description
0	Normal frame
1	Raised frame (3-D)
2	Inset frame (3-D)
3	Lowered frame (3-D)

Data Type

Integer (enumerated)

See Also

Properties:

    BevelInnerShape

    BevelOuter

    BevelOuterShape

    BevelWidth

    Border

    BorderWidth

BevelInnerShape Property

Description

Determines the shape of the 3-D style of the border immediately surrounding the LED area.

Usage

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

Remarks

The value of this property determines the shape of the LED's border.   If the Border property is set to False, this property is ignored.   If BevelOuterShape is set to 1 (Ellipse), this property is automatically set to 1.

This property may be one of two values:

Value	Description
0	Rectangle
1	Ellipse

Data Type

Integer (enumerated)
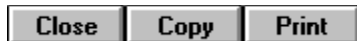
See Also

Properties:

[BevelInner](#)

[BevelOuter](#)

[BevelOuterShape](#)

[BevelWidth](#)

[Border](#)

[BorderWidth](#)

Bevel Example

In this example, the program will styles of the bevels used. To try this example, paste the code into the Declarations section of a form that contains a two horizontal scroll bars (one for BevelInner, and one for BevelOuter), two labels (which show the properties), and an LED control.   Press F5.   Play with   the scroll bars to see things change.

```
Sub Form_Load ()
    Me.BackColor = RGB(192, 192, 192)

    HScroll1.Value = 0
    HScroll1.Min = 0
    HScroll1.Max = 3

    HScroll2.Value = 0
    HScroll2.Min = 0
    HScroll2.Max = 3

    Call HScroll1_Change
    Call HScroll2_Change
End Sub

Sub HScroll1_Change ()
    LED1.BevelInner = HScroll2.Value
    Label1.Caption = "BevelInner:" & Format$( HScroll1.Value )
End Sub

Sub HScroll1_Scroll ()
    Call HScroll1_Change
End Sub

Sub HScroll2_Change ()
    LED1.BevelOuter = HScroll3.Value
    Label2.Caption = "BevelOuter:" & Format$( HScroll2.Value )
End Sub

Sub HScroll2_Scroll ()
    Call HScroll2_Change
End Sub
```

BevelOuter Property

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.   If the Border property is set to False, this property is ignored.   This property may be one of four values:

Value	Description
0	Normal frame
1	Raised frame (3-D)
2	Inset frame (3-D)
3	Lowered frame (3-D)

Data Type

Integer (enumerated)

See Also

Properties:

[BevelInner](#)
[BevelInnerShape](#)
[BevelOuterShape](#)
[BevelWidth](#)
[Border](#)
[BorderWidth](#)

BevelOuterShape Property

Description

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

Usage

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

Remarks

The value of this property determines the shape of the controls border.   If the <u>Border</u> property is set to False, this property is ignored.   If <u>BevelInnerShape</u> is set to 0 (Rectangle), this property is automatically set to 0 as well.

This property may be one of two values:

Value	Description
0	Rectangle
1	Ellipse

Data Type

Integer (enumerated)

See Also

Properties:

[BevelInner](#)
[BevelInnerShape](#)
[BevelOuter](#)
[BevelWidth](#)
[Border](#)
[BorderWidth](#)

BevelWidth Property

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, see Border and BevelOuter).   This is always measured in pixels.

Data Type

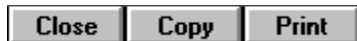Integer

See Also

Properties:
  [BevelInner](#)
  [BevelOuter](#)
  [Border](#)
  [BorderWidth](#)

Width Properties Example

In this example, the program will vary the width of bevels. To try this example, paste the code into the Declarations section of a form that contains two horizontal scroll bars, two labels, and an LED control. Press F5. Play with the scroll bars to see things change.

```
Sub Form_Load ()
    HScroll1.Value = 2
    HScroll1.Min = 0
    HScroll1.Max = 10

    HScroll2.Value = 2
    HScroll2.Min = 0
    HScroll2.Max = 10

    Call HScroll1_Change
    Call HScroll2_Change
End Sub

Sub HScroll1_Change ()
    LED1.BevelWidth = HScroll1.Value
    Label1.Caption = "BevelWidth:" & Format$( HScroll1.Value )
End Sub

Sub HScroll1_Scroll ()
    Call HScroll1_Change
End Sub

Sub HScroll2_Change ()
    LED1.BevelWidthBar = HScroll3.Value
    Label2.Caption = "BorderWidth:" & Format$( HScroll2.Value )
End Sub

Sub HScroll2_Scroll ()
    Call HScroll2_Change
End Sub
```

Border Property

Description

Determines whether or not there is a border around the control.

Usage

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

Remarks

The value of this property determines whether or not the control has a border.   This property may be one of two values:

Value	Description
0	No border
1	Single width border

Data Type

Integer (enumerated)

See Also

Properties:
BevelInner
BevelOuter

Border Property Example

In this example, the program will vary the width of the bars. To try this example, paste the code into the Declarations section of a form that contains a command button and an LED control.   Press F5.   Play with the check boxes to see things change.

```
Sub Form_Load ()
    Command1.Caption = "Border"
End Sub

Sub Command1_Click ()
    LED1.Border = 1 - LED1.Border

    If LED1.Border Then
        Command1.Caption = "Border"
    Else
        Command1.Caption = "No Border"
    End If
End Sub
```

BorderWidth Property

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 Border and BevelOuter) and the inner border (if any, see BevelInner).   This is always measured in pixels.

Data Type

Integer

See Also

Properties:

BevelInner

BevelOuter

Border

Value Property

Description

Turns the LED on or off.

Usage

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

Remarks

This property determines whether the LED is on or off.   If the control is disabled, the LED is painted using <u>DisabledColor</u>.   If Value is True, the LED is painted using <u>OnColor</u>.   And, if Value is False, the LED is painted using <u>OffColor</u>.

This property is the default value of the control.

Data Type

Integer (boolean)

See Also

Properties:

[DisabledColor](#)
[OffColor](#)
[OnColor](#)

Value Property Example

In this example, the program will show the percentage bar going up and down. To try this example, paste the code into the Declarations section of a form that contains a command button and an LED control.   Press F5.   Play with the button.

```
Sub Form_Load ()
    Me.BackColor = RGB( 192, 192, 192 )
    Command1.Caption = "Click Me"
End Sub

Sub Command1_Click ()
    LED1.Value = Not LED1.Value
End Sub
```

DisabledColor Property

Description

Sets the color of the LED when the control is disabled.

Usage

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

Remarks

This property determines what color the LED is when the control is disabled.

Data Type

Color

See Also

Properties:
OffColor
OnColor

OffColor Property

Description

Sets the color of the LED when the control is off.

Usage

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

Remarks

This property determines what color the LED is when the control is off (i.e., <u>Value</u> = False).

Data Type

Color

See Also

Properties:

    DisabledColor

    OnColor

    Value

OnColor Property

Description

Sets the color of the LED when the control is disabled.

Usage

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

Remarks

This property determines what color the LED is when the control is on (i.e., <u>Value</u> = True).

Data Type

Color

See Also

Properties:

DisabledColor

OffColor

Value

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 (*).

Click
DblClick
DragDrop
DragOver
MouseDown
MouseMove
MouseUp

Registration Information

Credits

LED was written by James Shields.   Inquiries can be sent to 71231,2066 on CompuServe, or mabry@halcyon.com on Internet.   Our mailing address is:

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

Registration

You can register this program by sending $15 ($20 for international orders) and your address.   You can register LED **and** its C source code by sending $35 ($40 for international orders).   With your order, you will receive a copy of our manual documenting all of our controls.

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

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).   LEDs SWREG ID number is 3380.   The source code version's ID number is 3381.

Credit Card Orders

We accept VISA and Mastercard.   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).

© Copyright 1994-1995 by Mabry Software, Inc.

Association of
Shareware
Professionals
Member

LED 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
BBS: WinDev BBS 206-634-0783
CompuServe: 71231,2066
Internet: mabry@halcyon.com

Where did you get this copy of LED?

_____

Ship to: _____

_____

_____

_____

_____

Phone: _____

Fax: _____

E-Mail: _____

MC/VISA: _____ exp. _____

Disk Size: (circle one)   3½   5¼

qty ordered ____   REGISTRATION
$15 each, postpaid (check or money order in hard currency). Outside
of North America add $5.00 shipping.

qty ordered ____   SOURCE CODE AND REGISTRATION
$35 each, postpaid (check or money order in hard currency). Outside
of North America add $5.00 shipping.

Getting Custom Controls Written

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

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

Phone: 206-634-1443
Fax: 206-632-0272
BBS: WinDev BBS 206-634-0783

CompuServe: 71231,2066
Internet: mabry@halcyon.com

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

Zane Thomas
Post Office Box 300
Indianola, WA   98342

CompuServe: 72060,3327