



## Help for LED

[Properties](#)

[Events](#)

### 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 (\*).

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

Value is the default value for the control.

## BevelInner Property

[See Also](#)

[Example](#)

### 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

[See Also](#)

### 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)

    HScroll11.Value = 0
    HScroll11.Min = 0
    HScroll11.Max = 3

    HScroll12.Value = 0
    HScroll12.Min = 0
    HScroll12.Max = 3

    Call HScroll11_Change
    Call HScroll12_Change
End Sub

Sub HScroll11_Change ()
    LED1.BevelInner = HScroll12.Value
    Label1.Caption = "BevelInner:" & Format$( HScroll11.Value )
End Sub

Sub HScroll11_Scroll ()
    Call HScroll11_Change
End Sub

Sub HScroll12_Change ()
    LED1.BevelOuter = HScroll13.Value
    Label2.Caption = "BevelOuter:" & Format$( HScroll12.Value )
End Sub

Sub HScroll12_Scroll ()
    Call HScroll12_Change
End Sub
```

## 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. 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

[See Also](#)

### 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 [Border](#) property is set to False, this property is ignored. If [BevelInnerShape](#) 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

[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, 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 ()
    HScroll11.Value = 2
    HScroll11.Min = 0
    HScroll11.Max = 10

    HScroll12.Value = 2
    HScroll12.Min = 0
    HScroll12.Max = 10

    Call HScroll11_Change
    Call HScroll12_Change
End Sub

Sub HScroll11_Change ()
    LED1.BevelWidth = HScroll11.Value
    Label1.Caption = "BevelWidth:" & Format$( HScroll11.Value )
End Sub

Sub HScroll11_Scroll ()
    Call HScroll11_Change
End Sub

Sub HScroll12_Change ()
    LED1.BevelWidthBar = HScroll13.Value
    Label2.Caption = "BorderWidth:" & Format$( HScroll12.Value )
End Sub

Sub HScroll12_Scroll ()
    Call HScroll12_Change
End Sub
```

## Border Property

[See Also](#)

[Example](#)

### 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

[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 [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

[See Also](#)

[Example](#)

### 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 [DisabledColor](#). If Value is True, the LED is painted using [OnColor](#). And, if Value is False, the LED is painted using [OffColor](#).

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

[See Also](#)

### Description

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

### Usage

*[form.]***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

[See Also](#)

### 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., Value = False).

### Data Type

Color

**See Also**

Properties:

[DisabledColor](#)

[OnColor](#)

[Value](#)

## OnColor Property

[See Also](#)

### 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., Value = 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

DbClick

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 order form 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). LED's 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.





## 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](mailto: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



