



## **ASSOCIATION OF SHAREWARE PROFESSIONALS (ASP) OMBUDSMAN STATEMENT**

**Global Majic Software, Inc.** is a member of the **Association of Shareware Professionals (ASP)**. ASP wants to make sure that the shareware principle works for you. If you are unable to resolve a shareware-related problem with **Global Majic Software, Inc.** by contacting them directly, ASP may be able to help. The ASP Ombudsman can help you resolve a dispute or problem with an ASP member, but does not provide technical support for members' products.

Please write to the ASP Ombudsman at:

545 Grover Road  
Muskegon, MI 49442-9427 USA  
FAX 616-788-2765

or send a CompuServe message via CompuServe Mail to:

ASP Ombudsman 70007,3536

# AUTOREDRAW PROPERTY

## Description

Determines whether the control is redraw manually or automatically.

## Usage

[*form.*]control.**AutoRedraw**[ = {TRUE|FALSE} ]

## Setting

The property settings are:

Setting	Description
<b>TRUE</b>	Automatic (default) - The operating system will redraw the control when it has time.
<b>FALSE</b>	Manual - The user is responsible for all redraw commands.

## Remarks

If AutoRedraw=**TRUE**, then the control will be redrawn after any property is changed. If several properties are being changed rapidly, then the control may seem slow and/or may not update when desired. In this case, it may be wise to set AutoRedraw=**FALSE** and issue a Redraw command after all the desired property changes are made.

## Related Property

Redraw

## Data Type

Integer (Boolean)

# AUTO SIZE PROPERTY

## Description

Enables the control to automatically size itself to fit its contents or to fit its contents into its size.

## Usage

[*form.*]control.**AutoSize**[ = {TRUE|FALSE} ]

## Setting

The AutoSize property settings are:

Setting	Description
0	None - width and height of the control can be set manually
1	Size Control - adjusts control size to fit LEDs
2	Size LEDs - adjusts LED size to fit in the control

## Data Type

Integer (Enumerated)

# BACKGROUNDPICTURE PROPERTY

## Description

Determines the background graphic to be displayed in the control.

## Usage

[*form.*]control.**BackgroundPicture**[ = *picture* ]

## Setting

The BackgroundPicture property settings are:

<b>Setting</b>	<b>Description</b>
<b>(none)</b>	No picture is displayed.
<b>(bitmap)</b>	At design-time, specify the bitmap file name to be displayed. At run-time, specify the bitmap using the <b>LoadPicture</b> ( or comparable) function.

## Remarks

When setting the picture at design-time, the picture will be saved with the form, and will be compiled into the executable. When the BackgroundPicture property is set to (none), the BackColor property is used to fill the background of the control.

## Data Type

Picture

# BEVELINNER PROPERTY

# BEVELOUTER PROPERTY

## Description

Sets or returns the inner or outer shadow styles of the control.

## Usage

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

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

## Setting

The property settings are:

Setting	Description
---------	-------------

---

0	None
---	------

1	Raised
---	--------

2	Inset
---	-------

## Remarks

These properties have no affect when BevelWidth=0.

## Related Properties

BevelWidth and BorderWidth

## Data Type

Integer (Enumerated)

## BEVELWIDTH PROPERTY

### Description

Sets or returns the shadow sizes of the inner and outer bevels of the control.

### Usage

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

### Related Properties

BevelInner, BevelOuter and BorderWidth

### Data Type

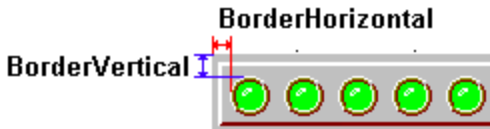
Integer

## BORDERHORIZONTAL PROPERTY

## BORDERVERTICAL PROPERTY

### Description

Sets or returns the horizontal or vertical border size of the control. The `BorderHorizontal` property defines the distance between the left edge of the control and the position of the upper left portion of the first LED. The `BorderVertical` property defines the distance between the top edge of the control and the position of the upper left portion of the first LED. These properties are clearly shown below



### Usage

`[form.]control.BorderHorizontal[ = integer ]`

`[form.]control.BorderVertical[ = integer ]`

### Data Type

Integer

## **BORDERWIDTH PROPERTY**

### **Description**

Sets or returns the border size between the inner and outer bevels of the control.

### **Usage**

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

### **Related Properties**

BevelInner, BevelOuter and BevelWidth

### **Data Type**

Integer





## Global Majic Software, Inc.



### LED Control

[Properties](#)

[Events](#)

[Product Support](#)

[Copyright](#)

#### Description:

This control displays a variety of LED styles including rectangular, circular, and user supplied bitmaps. The control can display a single LED as an on/off indicator or it may be configured as an array of horizontal or vertical LEDs to indicate an array of on/off indicators (Bitwise Mode). Additionally, it may be configured as a gauge or linear meter (Value Mode). The mouse can be used for input when the MouseControl property is set to **TRUE**. The control include bevels for a 3D appearance.

## DECAYRATE PROPERTY

### Description

Determines the rate (in milliseconds) at which the maximum marker decays.

### Usage

`[form.]control.DecayRate[ = integer ]`

### Remarks

This property only applies when MaxDecay = **TRUE**.

### Related Property

MaxDecay

### Data Type

Integer

# DIRECTION PROPERTY

## Description

Selects either a forward or backward fill for the LEDs.

## Usage

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

## Setting

The Direction property settings are:

Setting	Description
0	Forward. This is left to right for horizontal <u>Orientation</u> and top to bottom for vertical <u>Orientation</u> .
1	Backward. This is right to left for horizontal <u>Orientation</u> bottom to top for vertical <u>Orientation</u> .

## Data Type

Integer (Enumerated)

**Events:**

Click  
GotFocus  
LostFocus  
MouseDown  
MouseMove  
MouseUp

## Properties:

<u>AutoRedraw</u>	<u>LEDColor</u>	<u>OffPictureRed</u>
<u>AutoSize</u>	<u>LEDHeight</u>	<u>OffPictureYellow</u>
<u>BackColor</u>	<u>LEDIndex</u>	<u>OnPictureGreen</u>
<u>BackgroundPicture</u>	<u>LEDSep</u>	<u>OnPictureRed</u>
<u>BevelInner</u>	<u>LEDStatus</u>	<u>OnPictureYellow</u>
<u>BevelOuter</u>	<u>LEDStyle</u>	<u>Orientation</u>
<u>BevelWidth</u>	<u>LEDWidth</u>	<u>Redraw</u>
<u>BorderHorizontal</u>	<u>Left</u>	<u>TabIndex</u>
<u>BorderVertical</u>	<u>MaxDecay</u>	<u>TabStop</u>
<u>BorderWidth</u>	<u>Mode</u>	<u>Tag</u>
<u>DecayRate</u>	<u>MouseControl</u>	<u>Top</u>
<u>Direction</u>	<u>Name</u>	<u>Value</u>
<u>Enabled</u>	<u>NumLEDs</u>	<u>Visible</u>
<u>Height</u>	<u>OffPictureGreen</u>	<u>Width</u>
<u>Index</u>		

# LEDCOLOR PROPERTY

## Description

Selects the color or determines which bitmap to display (if LEDStyle=2 - bitmap) for the LED currently selected by LEDIndex.

## Usage

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

## Setting

The LEDColor property settings are:

Setting	Description
0	Red or referencing <u>OnPictureRed/OffPictureRed</u> ( <u>LEDStyle=2</u> )
1	Yellow or referencing <u>OnPictureYellow/OffPictureYellow</u> ( <u>LEDStyle=2</u> )
2	Green or referencing <u>OnPictureGreen/OffPictureGreen</u> ( <u>LEDStyle=2</u> )

## Remarks

If LEDStyle=2 (bitmap) and no bitmaps have been defined, the LED will **NOT** be visible in the control. See the **example** for more information on setting LED properties.

## Related Properties

LEDHeight, LEDIndex, LEDSep, LEDStatus, LEDStyle, LEDWidth and NumLEDs

## Data Type

Integer (Enumerated)

## LEDINDEX PROPERTY

### Description

Used to assign properties to individual LEDs. Each LED has a unique LEDIndex whose valid value ranges from 0 to NumLEDs-1.

### Usage

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

### Remarks

See the example for more information on setting LED properties.

### Related Properties

LEDColor, LEDHeight, LEDSep, LEDStatus, LEDStyle, LEDWidth and NumLEDs

### Data Type

Integer

# LEDSTATUS PROPERTY

## Description

Selects the *ON* or *OFF* status of the LED currently selected by LEDIndex.

## Usage

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

## Setting

The LEDStatus property settings are:

Setting	Description
0	Off
1	On

## Remarks

Individual LEDs can be turned *ON* using the LEDStatus property only if the Mode property is set to "bitwise." See the example for more information on setting LED properties.

## Related Properties

LEDColor, LEDHeight, LEDIndex, LEDSep, LEDStyle, LEDWidth and NumLEDs

## Data Type

Integer (Enumerated)



# LEDSTYLE PROPERTY

## Description

Selects the display type for the LED currently selected by LEDIndex.

## Usage

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

## Setting

The LEDStyle property settings are:

Setting	Description
0	Rectangular
1	Circular
2	Bitmap (uses User supplied bitmaps for the LEDs)

## Related Properties

LEDColor, LEDHeight, LEDIndex, LEDSep, LEDStatus, LEDWidth and NumLEDs

## Data Type

Integer (Enumerated)

## LEDHEIGHT PROPERTY

## LEDWIDTH PROPERTY

## LEDSEP PROPERTY

### Description

Determines the dimensions of an LED and the spacing between multiple LEDs.

### Usage

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

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

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

### Remarks

The LEDWidth is the dimension between the left and right edges of the LED. The LEDHeight is the dimension between the top and bottom edges of the LED. If multiple LEDs reside in the control, then each LED is separated by the dimension specified in the LEDSep property.

### Related Properties

LEDColor, LEDIndex, LEDStatus, LEDStyle and NumLEDs

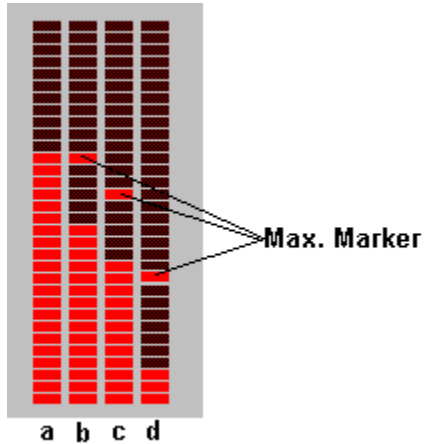
### Data Type

Integer

# MAXDECAY PROPERTY

## Description

Enables and disables the maximum marker for the LED control.



## Usage

`[form.]control.MaxDecay[ = {TRUE|FALSE} ]`

## Setting

The MaxDecay property settings are:

Setting	Description
TRUE	Enables the maximum marker.
FALSE	Disables the maximum marker.

## Related Property

[DecayRate.](#)

## Data Type

Integer (Boolean)

# MODE PROPERTY

## Description

Selects the analysis method of the Value property.

## Usage

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

## Setting

The Mode property settings are:

Setting	Description
0	The <u>Value</u> property determines the number of LEDs ON
1	Perform at bitwise analysis on the <u>Value</u> property to determine which LEDs are ON.

## Data Type

Integer (Enumerated)

# MOUSECONTROL PROPERTY

## Description

Enables or disables mouse input to the control.

## Usage

`[form.]control.MouseControl[ = {TRUE|FALSE} ]`

## Setting

The MouseControl property settings are:

Setting	Description
True	Allows the control's <u>Value</u> to be modified with mouse input.
False	Disables mouse input to the control.

## Data Type

Integer (Boolean)

## NUMLEDS PROPERTY

### Description

Defines the number of LEDs that appear in the control.

### Usage

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

### Remarks

See the [example](#) for more information on setting LED properties.

### Related Properties

[LEDColor](#), [LEDHeight](#), [LEDIndex](#), [LEDSep](#), [LEDStatus](#), [LEDStyle](#) and [LEDWidth](#)

### Data Type

Integer

**OFFPICTUREGREEN PROPERTY**  
**OFFPICTUREYELLOW PROPERTY**  
**OFFPICTURERED PROPERTY**  
**ONPICTUREGREEN PROPERTY**  
**ONPICTUREYELLOW PROPERTY**  
**ONPICTURERED PROPERTY**

### Description

Determines the graphic to be displayed in the *ON* and *OFF* positions when LEDStyle = 2 (bitmap). Although a color is referenced in the name of the property, it in no way limits the color scheme of the bitmap. It merely allows the bitmaps to be selected by the LEDColor property.

### Usage

```
[form.]control.OffPictureGreen[ = picture ]  
[form.]control.OffPictureYellow[ = picture ]  
[form.]control.OffPictureRed[ = picture ]  
[form.]control.OnPictureGreen[ = picture ]  
[form.]control.OnPictureYellow[ = picture ]  
[form.]control.OnPictureRed[ = picture ]
```

### Setting

The property settings are:

<b>Setting</b>	<b>Description</b>
<b>(none)</b>	No picture is displayed.
<b>(bitmap)</b>	At design-time, specify the bitmap file name to be displayed. At run-time, specify the bitmap using the <b>LoadPicture</b> (or comparable) function.

### Remarks

When setting the picture at design-time, the picture will be saved with the form and will be compiled into the executable.

### Data Type

Picture

# ORIENTATION PROPERTY

## Description

Determines whether the control is displayed horizontally or vertically.

## Usage

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

## Setting

The property settings are:

Setting	Description
0	Horizontal
1	Vertical

## Remarks

The Direction property may be used to alter the fill direction of the control.

## Data Type

Integer



## PRODUCT SUPPORT

Product support for all products is available to registered users by contacting **Global Majic Software, Inc.** at any of the following locations:

**CompuServe:** 73261,3642

**AmericaOnline:** GMagic

**Internet:** gms@globalmajic.com

**Snail Mail:** Global Majic Software, Inc.  
P.O. Box 322  
Madison, Alabama 35758

**TEL/FAX:** (205) 864-0708

**Home Page:** <http://www.globalmajic.com>

Product Support is free for a period of three (3) months from the date of registration.

If you have a shareware-related problem or dispute that you are unable to resolve with **Global Majic Software, Inc.**, please feel free to contact the [Association of Shareware Professionals](#).

# REDRAW PROPERTY

## Description

Issues a redraw command to the control if AutoRedraw=**FALSE**.

## Usage

[*form.*]control.Redraw[ = {TRUE|FALSE} ]

## Setting

The property settings are:

Setting	Description
TRUE	Issue a redraw command.
FALSE	Does not issue redraw command.

## Remarks

If AutoRedraw=**TRUE**, then the control will be redrawn after any property is changed. If several properties are being changed rapidly, then the control may seem slow and/or may not update when desired. In this case, it may be wise to set AutoRedraw=**FALSE** and issue a Redraw command after all the desired property changes are made.

## Related Property

AutoRedraw

## Data Type

Integer (Boolean)

# VALUE PROPERTY

## Description

Sets or returns the control's displayed value.

## Usage

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

## Remarks

This property has two possible uses, depending on the Mode property setting. If Mode = 0, then the value corresponds to the number of LEDs turned *ON*. However, if Mode = 1, then a bitwise analysis of the value is used to set the LEDs LEDStatus property. The value may also be returned by setting the LEDStatus property of individual LEDs by using the mouse.

## Data Type

Single

## COPYRIGHT INFORMATION

All **Global Majic Software, Inc.** software programs, shareware, and freeware are protected under the copyright laws of the United States and foreign countries. All rights are reserved to **Global Majic Software, Inc.** Violations of copyright laws are investigated by the FBI. Distribution of **Global Majic Software, Inc.** products implies that you have read and agreed to the distribution terms described below:

## INTENT

**Global Majic Software, Inc.** seeks to distribute its shareware as widely as possible. However, we want the end-users of our software to be properly informed that it is shareware.

## DISTRIBUTOR INFORMATION AND LICENSE INFORMATION

The license information and distribution requirements in this document supersede all previous license statements. To continue to distribute **Global Majic Software, Inc.** products, you must adhere to the licensing and distribution requirements below.

If you are a mail order or BBS-type distributor of shareware software, you may distribute these programs as they are, without any changes other than expanding files contained in the ZIP archives. However, you have the responsibility to check from time to time, at a minimum interval of 6 months, for new versions of these programs, and to update your copies in a timely manner. **Global Majic Software, Inc.** will gladly send you a diskette containing the current versions on request.

You must fully identify all **Global Majic Software, Inc.** programs in your advertising, by the program's full name and version, and indicate the registration fee in the program description. The words **Global Majic Software, Inc.** must appear in all program descriptions.

## SHAREWARE DISCLOSURE REQUIRED

All advertising and packaging information including references to **Global Majic Software, Inc.** products must contain a statement explaining the shareware concept. Specifically, that statement must explain that shareware software **MUST** be registered by the user, after a trial period, by paying a registration fee, and that all monies paid for the shareware version are duplication and distribution charges only. All such statements must be clearly displayed in a position where they are likely to be read by potential customers.

## RETAIL RACK AND CD-ROM DISTRIBUTION

If you distribute shareware in a retail setting in racks, store displays, vending machines, at computer fairs, or in any way other than normal BBS or catalog-based sales, you must contact **Global Majic Software, Inc.** for permission to distribute any **Global Majic Software, Inc.** program. Rack or retail-like sales require a special distribution license, normally requiring royalties paid to **Global Majic Software, Inc.** If you distribute shareware on CD-ROM disks, you must also contact **Global Majic Software, Inc.** before including any **Global Majic Software, Inc.** shareware programs on a CD-ROM disk. Normally, permission is granted, but current versions must be included and all old versions of any **Global Majic Software, Inc.** program removed from any CD-ROM disk containing **Global Majic Software, Inc.** products.

## EXAMPLE - HOW TO SETUP A LED CONTROL

### General Information

This example will explain the process of building a LED control. The final product is shown in Figure 1. Although this example seems strange, it does explain how several of the control's properties relate to each other. For example, it depicts how to use the properties to setup multiple LEDs to obtain a desired appearance. In general, the steps for setting up LEDs are as follows: **1)** set the LEDs "global" properties (NumLEDs, LEDStyle, etc.); **2)** set the unique id for the LED (LED1.LEDIndex=0); **3)** set the other LED properties (LEDColor, LEDStatus, etc); and **4)** change LEDIndex and repeat step 3 if desired. For simplicity, the steps for setting up this example are written out in Visual Basic script. However, the values may also be assigned using the property list at design time.



Figure 1

### Sample Code

```
LED1.NumLEDs = 10
LED1.LEDSep = 5
LED1.LEDStyle = 1

For i% = 0 To 9
    LED1.LEDIndex = i%
    Select Case i%
        Case 0, 1, 2, 3, 4, 5
            LED1.LEDColor = 2
        Case 6, 7
            LED1.LEDColor = 1
        Case 8, 9
            LED1.LEDColor = 0
    End Select
Next
LED1.LEDIndex = 3
LED1.LEDStatus = 1
LED1.LEDIndex = 7
LED1.LEDStatus = 1
LED1.LEDIndex = 8
LED1.LEDStatus = 1
```



