



## **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

## AUTOALIGN PROPERTY

### Description

Determines whether the selection caption alignment is controlled manually or automatically. If **AutoAlign=TRUE** then the control will automatically right justify all selections on the left of the control, left justify all selections on the right of the control, and center all selections at the top and bottom of the control. If, however, **AutoAlign=FALSE**, the user must enter the desired justification for each individual selection using the [SelectionAlign](#) property.

### Usage

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

### Remarks

For switches with a large number of selections, it may be advantageous to set **AutoAlign=TRUE**.

### Related Properties

[AutoAngle](#), [AutoAngleConfine](#), [AutoOffset](#), [AutoOffsetDistance](#), [AutoOffsetStyle](#), [AutoRadius](#), [AutoStartAngle](#) and [AutoStopAngle](#)

### Data Type

Integer (Boolean)

## AUTOANGLE PROPERTY

### Description

Determines whether the selection angles are controlled manually or automatically. If **AutoAngle=TRUE** then the control will evenly space the selected number of Selections between the angular extremes of the control as defined by the AutoStartAngle and AutoStopAngle properties. If, however, **AutoAngle=FALSE**, the user must enter the desired angle for each individual selection using the SelectionID and SelectionAngle properties.

### Usage

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

### Remarks

For switches with a large number of selections, it may be advantageous to set **AutoAngle=TRUE**.

### Related Properties

AutoAlign, AutoAngleConfine, AutoOffset, AutoOffsetDistance, AutoOffsetStyle, AutoRadius, AutoStartAngle and AutoStopAngle

### Data Type

Integer (Boolean)

## AUTOANGLECONFINE PROPERTY

### Description

Determines if the knob can be rotated outside of the extents defined by [AutoStartAngle](#) and [AutoStopAngle](#).

### Usage

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

### Related Properties

[AutoAlign](#), [AutoAngle](#), [AutoOffset](#), [AutoOffsetDistance](#), [AutoOffsetStyle](#), [AutoRadius](#), [AutoStartAngle](#) and [AutoStopAngle](#)

### Data Type

Integer (Boolean)

# AUTOOFFSET PROPERTY

## Description

Determines whether the selection caption offset is controlled manually or automatically. If **AutoOffset=TRUE** then the control will use the AutoOffsetStyle and AutoOffsetDistance properties to draw the control. If, however, **AutoOffset=FALSE**, the user must enter the desired offsets for each individual selection using the SelectionID, SelectionOffsetX and SelectionOffsetY properties.

## Usage

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

## Remarks

For switches with a large number of selections, it may be advantageous to set **AutoOffset=TRUE**.

## Related Properties

AutoAlign, AutoAngle, AutoAngleConfine, AutoOffsetDistance, AutoOffsetStyle, AutoRadius, AutoStartAngle and AutoStopAngle

## Data Type

Integer (Boolean)

# AUTOOFFSETDISTANCE PROPERTY

## Description

Determines the length of the offset (from the selector's center) when AutoOffsetStyle is not zero. This property is based on a unitless scale and typically has values between 0.0 and 1.0. If the AutoOffsetStyle=0 then this property has no effect on the control. If AutoOffsetStyle is not zero, however, than a line is drawn either horizontally or vertically using the appropriate distance.

## Usage

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

## Remarks

This property is ignored if AutoOffset=FALSE.

## Related Properties

AutoAlign, AutoAngle, AutoAngleConfine, AutoOffset, AutoOffsetStyle, AutoRadius, AutoStartAngle and AutoStopAngle

## Data Type

Single

# AUTOOFFSETSTYLE PROPERTY

## Description

Sets the type of offset for all selections.

## Usage

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

## Setting

The AutoOffsetStyle property settings are:

Setting	Description
0	Circular - no offset ( <u>AutoOffsetDistance</u> ignored)
1	Horizontal
2	Vertical

## Related Properties

AutoAlign, AutoAngle, AutoAngleConfine, AutoOffset, AutoOffsetDistance, AutoRadius, AutoStartAngle and AutoStopAngle

## Data Type

Integer (Enumerated)

# AUTORADIUS PROPERTY

## Description

Determines the radius (from the center of the knob) at which the labels are displayed. This property is based on a unitless scale and typically has values between 0.0 and 1.0. Additionally, the AutoOffsetDistance (if AutoOffset=TRUE) may be used to modify the label location in order to increase readability or for purely cosmetic reasons.

## Usage

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

## Remarks

The value of this property should be greater than zero.

## Related Properties

AutoAlign, AutoAngle, AutoAngleConfine, AutoOffset, AutoOffsetDistance, AutoOffsetStyle, AutoStartAngle and AutoStopAngle

## Data Type

Single



# **AUTOSTARTANGLE PROPERTY**

# **AUTOSTOPANGLE PROPERTY**

## **Description**

Determines the angular extremes of the selector switch. These properties are ignored when AutoAngle=FALSE.

## **Usage**

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

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

## **Remarks**

Start and stop angles are bound between 0 and 360 degrees. Values outside this range are automatically corrected.

## **Related Properties**

AutoAlign, AutoAngle, AutoAngleConfine, AutoOffset, AutoOffsetDistance, AutoOffsetStyle and AutoRadius

## **Data Type**

Single

## BACKGROUND COLOR PROPERTY

### Description

Determines the background color of the control. It is ignored if BackgroundPicture is set.

### Usage

`[form.]control.BackgroundColor[ = color ]`

### Remarks

This property can be set using Visual Basic's **RGB** or **QBColor** (or comparable) functions.

### Data Type

Long

# BACKGROUNDPICTURE PROPERTY

## Description

Determines the graphic to be displayed in the background of 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 Visual Basic's <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

# 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

# BORDERTYPE PROPERTY

## Description

Sets or returns the style of the border around the control.

## Usage

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

## Setting

The BoderType property settings are:

<b>Setting</b>	<b>Description</b>
<b>0</b> (None)	Border is not displayed.
<b>1</b> (Bevel)	3D beveled border is displayed using <u>BevelInner</u> , <u>BevelOuter</u> , <u>BevelWidth</u> and <u>BorderWidth</u> properties.
<b>2</b> (Outline)	Frame style border is displayed using <u>OutlineAlign</u> , <u>OutlineColor</u> , <u>OutlineTitle</u> and <u>OutlineWidth</u> properties.

## Data Type

Integer (Enumerated)

## **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

## CAPTION PROPERTY

### Description

Determines the text displayed on the control for the caption currently selected by CaptionID. The number of captions displayed is set by the Captions property.

### Usage

`[form.]control.Caption[ = string ]`

### Remarks

See the example for more information on setting caption properties.

### Related Properties

CaptionColor, CaptionFontID, CaptionID, Captions, CaptionX and CaptionY

### Data Type

String



## CAPTIONCOLOR PROPERTY

### Description

Determines the text color for the caption currently selected by [CaptionID](#).

### Usage

`[form.]control.CaptionColor[ = color ]`

### Remarks

This property can be set using Visual Basic's **RGB** or **QBColor** (or comparable) functions. See the [example](#) for more information on setting caption properties.

### Related Properties

[Caption](#), [CaptionFontID](#), [CaptionID](#), [Captions](#), [CaptionX](#) and [CaptionY](#)

### Data Type

Long

## CAPTIONFONTID PROPERTY

### Description

Determines which font (designated by FontID) is used for the caption currently selected by CaptionID.

### Usage

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

### Remarks

See the example for more information on setting caption properties.

### Related Properties

Caption, CaptionColor, CaptionID, Captions, CaptionX and CaptionY

### Data Type

Integer

# CAPTIONID PROPERTY

## Description

Assigns a unique ID to each caption. This property must be set before any other caption property (except Captions). The total number of captions is determined by the Captions property and CaptionID has valid values from 0 to Captions-1.

## Usage

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

## Remarks

The number of Captions must be set before this property can be set. See the **example** for more information on setting caption properties.

## Related Properties

Caption, CaptionColor, CaptionFontID, Captions, CaptionX and CaptionY

## Data Type

Integer

# CAPTIONX PROPERTY

# CAPTIONY PROPERTY

## Description

Determines the vertical and horizontal position of the caption currently selected by the CaptionID property. These properties are based on a unitless scale and typically have values between -1.0 and 1.0 where a value of 0.0 is located at the center of the control.

## Usage

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

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

## Remarks

See the example for more information on setting caption properties.

## Related Properties

Caption, CaptionColor, CaptionFontID, CaptionID and Captions

## Data Type

Single

## CAPTIONS PROPERTY

### Description

Determines the number of captions displayed on the control. This property must be set before all other caption properties are entered (see [example](#)). The [CaptionID](#) property is used to select the caption to which caption properties apply.

### Usage

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

### Remarks

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

### Related Properties

[Caption](#), [CaptionColor](#), [CaptionFontID](#), [CaptionID](#), [CaptionX](#) and [CaptionY](#)

### Data Type

Integer

A **CHANGE** event is fired every time the left button is released when using the mouse to change the value of the control.



## Global Majic Software, Inc.



### Selector Control

[Properties](#)

[Events](#)

[Product Support](#)

[Copyright](#)

#### Description:

This control is a highly customizable selector switch. Properties are provided to modify the knob and mark styles, selections, line, offsets, captions, border and background. The mouse may optionally be used to change selector settings.

#### Selections:

Each switch may have several possible selections. The caption, color, alignment, and offset of each selection is controlled by the user. Additionally, there are automatic features that define the control with minimal work required.

#### Lines and Offsets:

The control's lines are controlled by the user. Properties are available to change the line's inner radius and thickness as well as whether or not lines are visible. Offsets, on the other hand, may be controlled automatically or the user. The length and angle of each offset can be set.

#### Knob:

There are several styles of knobs that may be used in the selector switch control. Properties are provided to modify the knob's style, inner-outer radii, width, and color. Additionally, user defined shapes may be specified.

#### Mark:

Each knob may be adorned with a single mark. Properties are provided to modify the mark's style, inner-outer radii, width, and color. Additionally, user defined mark shapes may be incorporated into the control.

#### Captions:

The control can be embellished with multiple captions to indicate the type of measurement being displayed, units used or any other informative or decorative labeling. These captions are in no way related to the captions defined for each selection.

## 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.



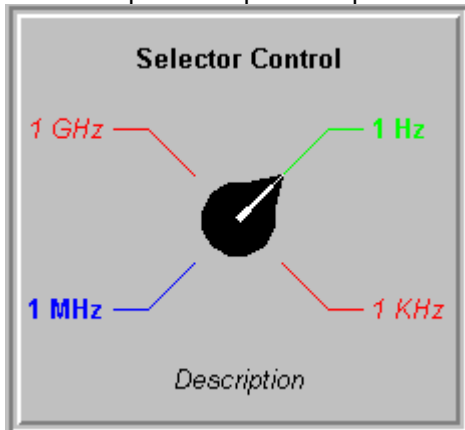
**Events:**

Change  
Click  
GotFocus  
KeyDown  
KeyPress  
KeyUp  
LostFocus  
MouseDown  
MouseMove  
MouseUp

## EXAMPLE - HOW TO BUILD A SELECTOR

### General Information

This example will explain the process of building a selector knob. The final product is shown



The purpose of this example is to show how several of the control's properties relate to each other. For example, it depicts how to use the selection properties to obtain the desired appearance. In general, the steps for setting up selections are as follows: **1)** set the number of selections (`Selector1.Selections=2`); **2)** set the unique id for the tic set (`Selector1.SelectionID=0`); **3)** set the other selection properties (`Selector1.SelectionAngle=45`); and **4)** change `SelectionID` and repeat step 3 if desired. This same process is used to set the properties for all the indexed items contained in the control (selections, captions, etc.). 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.

### Sample Code

#### 'general setup

```
Selector1.AutoAngle = False
Selector1.AutoOffsetDistance = 0.8
Selector1.Highlight = True
Selector1.HighlightColor = &HFF00&
Selector1.KnobColor = &H0&
Selector1.KnobStyle = 1
Selector1.KnobRadius = 0.4
Selector1.MarkStyle = 2
Selector1.MarkColor = &HFFFFFF
Selector1.MarkOuterRadius = 1
Selector1.MarkWidth = 0.3
```

#### 'setup fonts

```
Selector1.Fonts = 2
```

```
Selector1.FontID = 0
Selector1.FontBold = True
Selector1.FontName = "Arial"
Selector1.FontSize = 12
```

```
Selector1.FontID = 1
Selector1.FontBold = False
Selector1.FontItalic = True
Selector1.FontName = "Arial"
Selector1.FontSize = 12
```

#### 'setup captions

```
Selector1.Captions = 2
```

Selector1.CaptionID = 0  
Selector1.CaptionFontID = 0  
Selector1.Caption = "Selector Control"  
Selector1.CaptionX = 0  
Selector1.CaptionY = 1

Selector1.CaptionID = 1  
Selector1.CaptionFontID = 1  
Selector1.Caption = "Description"  
Selector1.CaptionX = 0  
Selector1.CaptionY = -1

### **'setup selections**

Selector1.Selections = 4  
Selector1.SelectionID = 0  
Selector1.SelectionFontID = 0  
Selector1.SelectionCaption = " 1 Hz "  
Selector1.SelectionAngle = 45  
Selector1.SelectionColor = &HFF0000

Selector1.SelectionID = 1  
Selector1.SelectionFontID = 1  
Selector1.SelectionCaption = " 1 KHz "  
Selector1.SelectionAngle = 135  
Selector1.SelectionColor = &HFF&

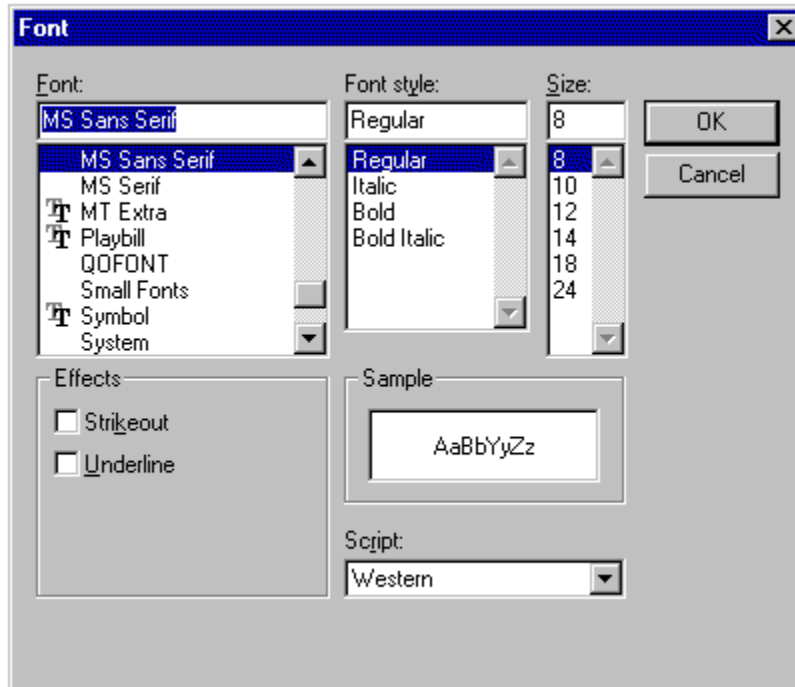
Selector1.SelectionID = 2  
Selector1.SelectionFontID = 0  
Selector1.SelectionCaption = " 1 MHz "  
Selector1.SelectionAngle = -135  
Selector1.SelectionColor = &HFF0000

Selector1.SelectionID = 3  
Selector1.SelectionFontID = 1  
Selector1.SelectionCaption = " 1 GHz "  
Selector1.SelectionAngle = -45  
Selector1.SelectionColor = &HFF&

## FONTDIALOG PROPERTY

### Description

Selecting this property (in design mode) launches the font dialog box shown below. This dialog sets the font properties for the font currently selected by FontID.



### Usage

This property can only be used at design time. Use standard font properties to set fonts in code.

### Related Properties

FontBold, FontID, FontItalic, FontName, Fonts, FontSize, FontStrike and FontUnder

### Data Type

N/A

# FONTID PROPERTY

## Description

This property allows the control to display several different fonts by assigning a unique ID to each font. The total number of fonts is determined by the Fonts property and FontID has valid values from 0 to Fonts-1.

## Usage

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

## Remarks

The desired font is obtained by selecting the corresponding FontID (through the use of CaptionFontID for example). See the example for more information on setting font properties.

## Related Properties

CaptionFontID, FontBold, FontDialog, FontItalic, FontName, Fonts, FontSize, FontStrike, FontUnder and SelectionFontID

## Data Type

Integer

## FONTS PROPERTY

### Description

Determines the number of fonts displayed on the control. This property must be set before all other font properties are entered (see [example](#)). The [FontID](#) property is used to select the font to which other font properties apply.

### Usage

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

### Remarks

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

### Related Properties

[CaptionFontID](#), [FontBold](#), [FontDialog](#), [FontID](#), [FontItalic](#), [FontName](#), [FontSize](#), [FontStrike](#), [FontUnder](#) and [SelectionFontID](#)

### Data Type

Integer

## FRAMECOLOR PROPERTY

### Description

Determines the frame color surrounding the control's face.

### Usage

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

### Remarks

This property only applies when FrameStyle is set and FramePicture has not been specified. This property can be set using Visual Basic's **RGB** or **QBColor** (or comparable) functions.

### Related Properties

FramePicture, FrameScaleX, FrameScaleY and FrameStyle

### Data Type

Long

# FRAMEPICTURE PROPERTY

## Description

Determines the graphic to be displayed in the frame surrounding the control's face.

## Usage

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

## Setting

The FramePicture property settings are:

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

## Remarks

This property only applies when [FrameStyle](#) is set. When setting the picture at design-time, the picture will be saved with the form and will be compiled into the executable.

## Related Properties

[FrameColor](#), [FrameScaleX](#), [FrameScaleY](#) and [FrameStyle](#)

## Data Type

Picture



# FRAMESCALEX PROPERTY

# FRAMESCALEY PROPERTY

## Description

Determines the vertical and horizontal size of the opening inside of the frame surrounding the control. For a circular [FrameStyle](#), `FrameScaleY` is ignored and `FrameScaleX` is used to define the radius of the opening inside the frame surrounding the control. This property is based on a [unitless scale](#) and typically has values between 0.0 and 1.0.

## Usage

`[form.]control.FrameScaleX[ = single ]`

`[form.]control.FrameScaleY[ = single ]`

## Related Properties

[FrameColor](#), [FramePicture](#) and [FrameStyle](#)

## Data Type

Single

# FRAMESTYLE PROPERTY

## Description

Sets or returns the style of the frame surrounding the control.

## Usage

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

## Setting

The FrameStyle property settings are:

Setting	Description
0 (None)	A frame is not displayed.
1 (Circle)	A circular frame is displayed using <u>FrameScaleX</u> as the internal radius.
2 (Rectangle)	A rectangular frame is displayed using <u>FrameScaleX</u> and <u>FrameScaleY</u> properties to size the interior size of the frame.

## Related Properties

FrameColor, FramePicture, FrameScaleX and FrameScaleY

## Data Type

Integer (Enumerated)

## HIGHLIGHT PROPERTY

### Description

Determines whether the a selection's original color is altered to HighlightColor when it is selected.

### Usage

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

### Remarks

This property is applicable in both manual and automatic mode.

### Related Property

HighlightColor

### Data Type

Integer (Boolean)

# HIGHLIGHTCOLOR PROPERTY

## Description

Determines the color of the currently designated selection's line and caption.

## Usage

`[form.]control.HighlightColor[ = color ]`

## Remarks

This property can be set using Visual Basic's **RGB** or **QBColor** (or comparable) functions.

## Related Properties

[Highlight](#)

## Data Type

Long

# **KNOB COLOR PROPERTY**

## **Description**

Determines the knob's color.

## **Usage**

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

## **Remarks**

This property can be set using Visual Basic's **RGB** or **QBColor** (or comparable) functions.

## **Related Properties**

[KnobOffsetX](#), [KnobOffsetY](#), [KnobRadius](#), [KnobStyle](#), [KnobUserDefined](#) and [Value](#)

## **Data Type**

Long

# **KnobOriginX Property**

# **KnobOriginY Property**

## **Description**

Determines the vertical and horizontal positions of the knob's center. These properties are based on a unitless scale and typically have values between -1.0 and 1.0, where a value of 0.0 is located in the center of the control.

## **Usage**

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

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

## **Related Properties**

KnobColor, KnobRadius, KnobStyle, KnobUserDefined, and Value

## **Data Type**

Single

# **KNOBRADIUS PROPERTY**

## **Description**

Determines the outer radius of the knob. This property is based on a unitless scale and typically has values between 0.0 and 1.0.

## **Usage**

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

## **Remarks**

The outer radius should be greater than zero.

## **Related Properties**

KnobColor, KnobOriginX, KnobOriginY, KnobStyle, KnobUserDefined, and Value

## **Data Type**

Single

# KNOBSTYLE PROPERTY

## Description

Sets or returns the style of the knob.

## Usage

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

## Setting

The KnobStyle property settings are:

Setting	Description
0	Circular
1	Pointer
2	Complex
3	User-defined

## Related Properties

KnobColor, KnobOriginX, KnobOriginY, KnobRadius, KnobUserDefined, and Value

## Data Type

Integer (Enumerated)



# KNOBUSERDEFINED PROPERTY

# MARKUSERDEFINED PROPERTY

## Description

Determines the shape of a user defined knob or mark. This property only applies when KnobStyle = 4 (for knobs) or when MarkStyle=4 (for marks). This property defines a polygon made from a list of x,y coordinate pairs (comma delimited). The shape is defined using a coordinate system in which (0,0) is located at the center of the knob and the "pointer" of the knob (or tip of the mark) is typically defined as (0,1000).

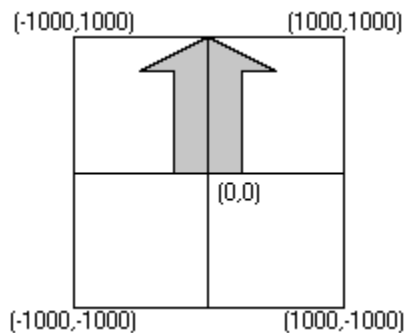
## Usage

[form.]control.KnobUserDefined[ = string ]  
[form.]control.MarkUserDefined[ = string ]

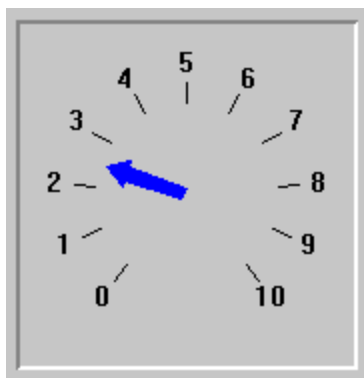
## Example

CNTL1.MarkUserDefined = "250,0, 250,750, 500,750, 0,1000, 00,750, -250,750, -250,0"

The following image displays the shape polygon defined in a coordinate system with origin at (0,0) and (1000,1000) extents.



The following image is a screen snap shot of the above shape used in a simple control (knob not shown).



## Related Properties

KnobUserDefined: [KnobColor](#), [KnobOriginX](#), [KnobOriginY](#), [KnobRadius](#), [KnobStyle](#) and [Value](#)

MarkUserDefined: [MarkColor](#), [MarkInnerRadius](#), [MarkOuterRadius](#), [MarkStyle](#) and [MarkWidth](#)

**Data Type**  
String

## LINEDISPLAY PROPERTY

### Description

Determines whether the lines to each selection are shown.

### Usage

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

### Remarks

This property is applicable in both manual and automatic modes.

### Related Properties

LineInnerRadius and LineThickness

### Data Type

Integer (Boolean)

## LINEINNERRADIUS PROPERTY

### Description

Determines the inner radius of the lines to each selection. This property is based on a unitless scale and typically has values between 0.0 and 1.0.

### Usage

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

### Remarks

The inner radius should be greater than zero.

### Related Properties

LineDisplay and LineThickness

### Data Type

Single

## LINE THICKNESS PROPERTY

### Description

Determines the thickness of the line(s) on the selector control. This property is measured in pixels.

### Usage

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

### Related Properties

[LineDisplay](#) and [LineInnerRadius](#)

### Data Type

Integer

## MARKCOLOR PROPERTY

### Description

Determines the knob's mark color.

### Usage

`[form.]control.MarkColor[ = color ]`

### Remarks

This property can be set using Visual Basic's **RGB** or **QBColor** (or comparable) functions.

### Related Properties

[MarkInnerRadius](#), [MarkOuterRadius](#), [MarkStyle](#), [MarkUserDefined](#), and [MarkWidth](#)

### Data Type

Long

## **MARKINNERRADIUS PROPERTY**

## **MARKOUTERRADIUS PROPERTY**

### **Description**

Determines the inner and outer radii of the knob's mark. The scale of these properties is based on the knob's size. For example, an inner radius of 0.0 and an outer radius of 1.0 yields a mark from the knob's center to its outer edge.

### **Usage**

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

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

### **Remarks**

The inner radius should be less than the outer radius.

### **Related Properties**

[MarkColor](#), [MarkStyle](#), [MarkUserDefined](#), and [MarkWidth](#)

### **Data Type**

Single

# MARKSTYLE PROPERTY

## Description

Sets or returns the style of the knob's mark.

## Usage

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

## Setting

The MarkStyle property settings are:

Setting	Description
0	None
1	Circular
2	Triangular
3	Arrow
4	User-defined

## Related Properties

[MarkColor](#), [MarkInnerRadius](#), [MarkOuterRadius](#), [MarkUserDefined](#), and [MarkWidth](#)

## Data Type

Integer (Enumerated)



## MARKWIDTH PROPERTY

### Description

Determines the width of the knob's mark. This property is based on a unitless scale and typically has values between 0.0 and 1.0.

### Usage

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

### Related Properties

MarkColor, MarkInnerRadius, MarkOuterRadius, MarkStyle, and MarkUserDefined

### Data Type

Single

# MOUSECONTROL PROPERTY

## Description

Enables and disables mouse input to the control.

## Usage

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

## Setting

The MouseControl property settings are:

Setting	Description
<b>TRUE</b>	Allows the control's value to be modified with mouse input.
<b>FALSE</b>	Disables mouse input to the control.

## Data Type

Integer (Boolean)

# OUTLINEALIGN PROPERTY

## Description

Determines the alignment of the OutlineTitle at the top of the control. This property only applies when BorderType is set to 2 (Outline).

## Usage

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

## Setting

The OutlineAlign property settings are:

Setting	Description
0	Left
1	Right
2	Center

## Related Properties

BorderType, OutlineColor, OutlineTitle and OutlineWidth

## Data Type

Integer (Enumerated)

# OUTLINECOLOR PROPERTY

## Description

Determines the color of the outline frame (and OutlineTitle) surrounding the control when BorderType is set to 2 (Outline).

## Usage

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

## Remarks

This property can be set using Visual Basic's **RGB** or **QBColor** (or comparable) functions.

## Related Properties

BorderType, OutlineAlign, OutlineTitle and OutlineWidth

## Data Type

Long

## OUTLINETITLE PROPERTY

### Description

Determines the text displayed as the title in the outline frame surrounding the control when the BorderType is set to 2 (Outline).

### Usage

[*form.*]control.**OutlineTitle**[ = *string* ]

### Related Properties

BorderType, OutlineAlign, OutlineColor and OutlineWidth

### Data Type

String

## OUTLINEWIDTH PROPERTY

### Description

Determines the thickness (in pixels) of the outline frame surrounding the control when BorderType is set to 2 (Outline).

### Usage

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

### Related Properties

BorderType, OutlineAlign, OutlineColor and OutlineTitle

### 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](#).

## Properties:

<u>AutoAlign</u>	FontName	<u>MarkStyle</u>
<u>AutoAngle</u>	Fonts	<u>MarkUserDefined</u>
<u>AutoAngleConfine</u>	FontSize	<u>MarkWidth</u>
<u>AutoOffset</u>	FontStrike	<u>MouseControl</u>
<u>AutoOffsetDistance</u>	FontUnder	<u>MousePointer</u>
<u>AutoOffsetStyle</u>	<u>FrameColor</u>	Name
<u>AutoRadius</u>	<u>FramePicture</u>	<u>OutlineAlign</u>
<u>AutoStartAngle</u>	<u>FrameScaleX</u>	<u>OutlineColor</u>
<u>AutoStopAngle</u>	<u>FrameScaleY</u>	<u>OutlineTitle</u>
<u>BackgroundColor</u>	<u>FrameStyle</u>	<u>OutlineWidth</u>
<u>BackgroundPicture</u>	Height	<u>SelectionAlign</u>
<u>BevelInner</u>	<u>Highlight</u>	<u>SelectionAngle</u>
<u>BevelOuter</u>	<u>HighlightColor</u>	<u>SelectionCaption</u>
<u>BevelWidth</u>	Index	<u>SelectionColor</u>
<u>BorderType</u>	<u>KnobColor</u>	<u>SelectionFontID</u>
<u>BorderWidth</u>	<u>KnobOriginX</u>	<u>SelectionID</u>
<u>Caption</u>	<u>KnobOriginY</u>	<u>SelectionOffsetX</u>
<u>CaptionColor</u>	<u>KnobRadius</u>	<u>SelectionOffsetY</u>
<u>CaptionFontID</u>	<u>KnobStyle</u>	<u>SelectionRadius</u>
<u>CaptionID</u>	<u>KnobUserDefined</u>	<u>Selections</u>
<u>Captions</u>	Left	TabIndex
<u>CaptionX</u>	<u>LineDisplay</u>	TabStop
<u>CaptionY</u>	<u>LineInnerRadius</u>	Tag
Enabled	<u>LineThickness</u>	Top
FontBold	<u>MarkColor</u>	<u>Value</u>
FontDialog	<u>MarkInnerRadius</u>	Visible
<u>FontID</u>	<u>MarkOuterRadius</u>	Width
FontItalic		



# SELECTIONALIGN PROPERTY

## Description

Sets the justification of the selection's caption currently specified by [SelectionID](#).

## Usage

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

## Setting

The SelectionAlign property settings are:

<b>Setting</b>	<b>Description</b>
0	Left Justify
1	Right Justify
2	Center

## Remarks

This property is ignored if [AutoAlign](#)=TRUE. See the [example](#) for more information on setting selection properties.

## Related Properties

[SelectionAngle](#), [SelectionCaption](#), [SelectionColor](#), [SelectionFontID](#), [SelectionID](#), [SelectionOffsetX](#), [SelectionOffsetY](#), [SelectionRadius](#) and [Selections](#)

## Data Type

Integer (Enumerated)

## SELECTIONANGLE PROPERTY

### Description

Determines the angle (zero up, positive clockwise) of the selection currently specified by SelectionID. If AutoAngle=TRUE, this property is ignored.

### Usage

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

### Remarks

Angles greater than 360 or less than zero degrees are automatically corrected. See the example for more information on setting selection properties.

### Related Properties

SelectionAlign, SelectionCaption, SelectionColor, SelectionFontID, SelectionID, SelectionOffsetX, SelectionOffsetY, SelectionRadius and Selections

### Data Type

Single

## SELECTIONCAPTION PROPERTY

### Description

Determines the text displayed for the selection specified by SelectionID. The number of selections displayed on the control is set by the Selections property.

### Usage

[*form.*]control.**SelectionCaption**[ = *string* ]

### Remarks

See the example for more information on setting selection properties.

### Related Properties

SelectionAlign, SelectionAngle, SelectionColor, SelectionFontID, SelectionID, SelectionOffsetX, SelectionOffsetY, SelectionRadius and Selections

### Data Type

String

# SELECTIONCOLOR PROPERTY

## Description

Determines the color of the SelectionCaption (and its associated line) currently specified by SelectionID.

## Usage

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

## Remarks

This property can be set using Visual Basic's **RGB** or **QBColor** (or comparable) functions. See the example for more information on setting selection properties.

## Related Properties

SelectionAlign, SelectionAngle, SelectionCaption, SelectionFontID, SelectionID, SelectionOffsetX, SelectionOffsetY, SelectionRadius and Selections

## Data Type

Long

## SELECTIONFONTID PROPERTY

### Description

Determines which font (designated by FontID) is used in the caption for the selection currently specified by SelectionID.

### Usage

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

### Remarks

See the example for more information on setting selection properties.

### Related Properties

SelectionAlign, SelectionAngle, SelectionCaption, SelectionColor, SelectionID, SelectionOffsetX, SelectionOffsetY, SelectionRadius and Selections

### Data Type

Integer

# SELECTIONID PROPERTY

## Description

Assigns a unique ID to each selection. This property must be set before any other selection property (except Selections). The total number of selections is determined by the Selections property and SelectionID has valid values from 0 to Selections-1.

## Usage

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

## Remarks

The number of Selections must be set before this property can be set. See the **example** for more information on setting selection properties.

## Related Properties

SelectionAlign, SelectionAngle, SelectionCaption, SelectionColor, SelectionFontID, SelectionOffsetX, SelectionOffsetY, SelectionRadius and Selections

## Data Type

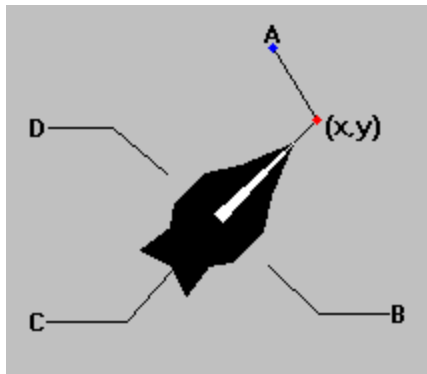
Integer

## SELECTIONOFFSETX PROPERTY

## SELECTIONOFFSETY PROPERTY

### Description

Defines an (x,y) pair that is used to determine the length of the offset (when AutoOffset=FALSE) of the current selection defined by SelectionID. If the LineDisplay=TRUE, then there is an (x,y) pair that is defined by SelectionAngle and SelectionRadius. This point is shown below as a red dot. The control uses SelectionOffsetX and SelectionOffsetY to define an additional (x,y) pair relative to the first pair (shown as a blue dot). These two points define the line that is drawn as the offset. These properties are based on a unitless scale and typically have values between -1.0 and 1.0.



### Usage

[form.]control.**SelectionOffsetX**[ = single ]  
[form.]control.**SelectionOffsetY**[ = single ]

### Remarks

The value of these properties should be greater than zero.

### Related Properties

SelectionAlign, SelectionAngle, SelectionCaption, SelectionColor, SelectionFontID, SelectionID, SelectionRadius and Selections

### Data Type

Single

# SELECTIONRADIUS PROPERTY

## Description

Determines the radius (from the center of the selector) at which the labels are displayed for the current selection (specified by SelectionID). This property is based on a unitless scale and typically has values between 0.0 and 1.0. Additionally, the SelectionOffsetX and SelectionOffsetY properties (if AutoOffset=FALSE) may be used to modify the label location in order to increase readability or for purely cosmetic reasons.

## Usage

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

## Remarks

The value of this property should be greater than zero. See the **example** for more information on setting selection properties.

## Related Properties

SelectionAlign, SelectionAngle, SelectionCaption, SelectionColor, SelectionFontID, SelectionID, SelectionOffsetX, SelectionOffsetY and Selections

## Data Type

Single



# SELECTIONS PROPERTY

## Description

Determines the number of selections displayed on the control. This property must be set before all other selection properties are entered (see [example](#)). The [SelectionID](#) property is used to choose the selection to which the selection properties apply.

## Usage

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

## Remarks

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

## Related Properties

[SelectionAlign](#), [SelectionAngle](#), [SelectionCaption](#), [SelectionColor](#), [SelectionFontID](#), [SelectionID](#), [SelectionOffsetX](#), [SelectionOffsetY](#) and [SelectionRadius](#)

## Data Type

Integer

# VALUE PROPERTY

## Description

Determines which selection to which the control points. The Value property has valid values from 0 to SelectionID-1.

## Usage

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

## Related Properties

Selections and SelectionID

## Data Type

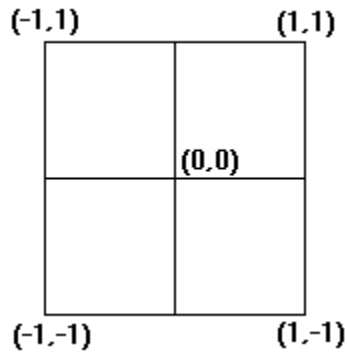
Integer

FontBold, FontItalic, FontName, FontSize, FontStrike and FontUnder

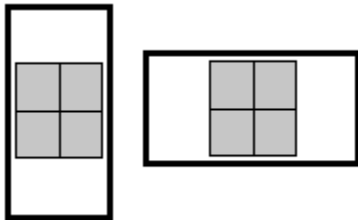
## UNITLESS SCALE

### Description:

Properties which are used to specify position or length use a scale which is based on the size of the control (instead of twips or pixels). For a control which is square, the coordinate system used is depicted with its origin at the center of the control and its width and height measured from -1 to 1.



For controls which are not square, the origin is still at the center of the control. The unit scale used, however, is based on the width or height whichever is smaller so that a unit square fits completely within the control as shown in the two controls below.



**NOTE:** For some variables (i.e., radii, width, etc.), the valid range of the unitless scale is from 0 to 1 (negative values have no meaning)



