Help for Knob

Properties Events Frequently Asked Questions

Registration Information

Order Form

Getting Custom Controls Written

Licensing Information

Description

The Knob VBX/OCX custom control displays a round knob that behaves like a slider or scroll bar.

Four different knob styles to choose from make Knob extremely versatile. A collection of flexible tickmark properties greatly enhances Knob's usefulness for developing commercial applications. Easily control knob color with five different color properties. Over thirty knob properties are at your fingertips!

- Normal, raised, lowered, and textured knob styles.
- User-definable number of tickmarks with captions.
- Full control of tick caption fonts.
- Control knob color, tick color, tick caption color, and background color.
- Bevel properties for 3D style look.

File Name

KNOB1.VBX, KNOB16.OCX, KNOB32.OCX

ActiveX / OCX Object Name

Mabry.KnobCtrl

ActiveX Compatility

VB 4.0 (32-bit) and 5.0

ActiveX Built With

Microsoft Visual C++ v4

ActiveX - Required DLLs

MFC40.DLL (October 6th, 1995 or later)
OLEPRO32.DLL (October 6th, 1995 or later)
MSVCRT40.DLL (September 29th, 1995 or later)

VBX Object Type

Knob

VBX Compatibility

VB 2.0, 3.0 and 4.0 (16-bit)

VBX Built With

Microsoft Visual C++ v1.5

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

Close

Knob Properties

Properties that have special meaning for this control or that only apply to this control are marked with an asterisk (*).

BackColor Property
*__**BevelWidth** Property__
*__**BorderWidth** Property__
Enabled Property
FontBold Property
FontItalic Property
FontName Property
FontSize Property
FontStrikethru Property
FontUnderline Property
Height Property
hWnd Property
Index Property
*__**Indicator** Property__
*__**IndicatorColor** Property__
*__**IndicatorWidth** Property__
*__**KnobColor** Property__
*__**KnobStyle** Property__
Left Property
*__**LinkControl** Property__
*__**LinkProperty** Property__
*__**Max** Property__
*__**Min** Property__
Name Property
Parent Property
*__**Radius** Property__
Tag Property
*__**TickCaption** Property__
*__**TickCaptionColor** Property__
*__**TickColor** Property__
*__**TickCount** Property__
*__**TickGap** Property__
*__**TickLength** Property__
*__**TickWidth** Property__
Top Property
*__**Value** Property__
*__**Version** Property__
Visible Property
Width Property

Close

Knob Events

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

Close

Frequently Asked Questions

MIDI Pack - General Questions

I am writing some Karaoke software and I need to play the MIDI file and show the lyric at the same time. If I put the lyric as a MARK or in a TRACK, can your software show me the lyric in some way?

I am writing some Karaoke software and I need to play the MIDI file and show the lyric at the same time. If I put the lyric as a MARK or in a TRACK, can your software show me the lyric in some way?

Frequently Asked Questions

The MIDI specification provides for that sort of application. Programming it is going to be a bit tricky, though. The basic approach would be to use the MIDIFILE control to find the strings and associated timing information, play the file with MCI, and keep track of where the playback is so that the messages could be displayed at the right times. Not an entirely trivial programming job.

Registration Information

CREDITS

Knob was written by James Shields.

CONTACT INFORMATION

Orders, inquiries, technical support, questions, comments, etc. can be sent to mabry@mabry.com on the Internet. Our mailing address/contact information is:

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

Sales: 1-800-99-MABRY (U.S. Only)
Voice: 206-634-1443
Fax: 206-632-0272 or 206-364-3196
Web: http://www.mabry.com

COST

The price of Knob (control only) is US$20 (US$25 for International orders). The cost of Knob and the C/C++ source code (of the control itself) is US$45 (US$50 for International orders).

Prices are subject to change without notice.

DELIVERY METHODS

We can ship this software to you via air mail and/or e-mail.

Air Mail - you will receive disks, a printed manual, and printed receipt if you choose this delivery method. The costs are:

US$5.00	US Priority Mail
US$10.00	AirBorne Express 2nd Day (US deliveries only)
US$15.00	AirBorne Express Overnight (US deliveries only)
US$45.00	International AirBorne Express.

E-Mail - We can ship this package to you via e-mail. You need to have an e-mail account that can accept large file attachments (which includes CompuServe, AOL, and most Internet providers). If you choose this option, please note: a printed manual is not included. We will, however, e-mail a receipt to you.

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 E-Mail - CompuServe members can use the software registration forum (GO SWREG) to register this package. Knob's SWREG ID number is 10293. The source code version's ID number is 10294. 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.

ORDER / PAYMENT METHODS

You can order this software by phone, fax, e-mail, mail. For your convenience, an order form has been provided that you can print out directly from this help file.

Please note that orders must include all information that is requested on our order form. Your shipment WILL BE DELAYED if we have to contact you for additional information (such as phone number, street address, etc.).

You can pay by credit card (VISA, MasterCard, American Express), check (U.S. dollars drawn on a U.S. bank), cash, International Money Order, International Postal Order, Purchase Order (established business entities only - terms net 30), or wire transfer.

WIRE TRANSFER INFORMATION

Here is the information you need regarding our account for a wire funds transfer:

Bank Name:	SeaFirst - Stone Way Branch
Bank Address:	3601 Stone Way North Seattle, WA 98103
Bank Phone:	206-585-4951
Account Name:	Mabry Software, Inc.
Routing Number:	12000024
Account Number:	16311706

If you are paying with a wire transfer of funds, please add US$12.50 to your order. This is the fee that SeaFirst Bank charges Mabry Software. Also, please ADD ANY ADDITIONAL FEES THAT YOUR BANK MAY CHARGE for wire transfer service. If you are paying with a wire transfer, we must have full payment deposited to our account before we can ship your order.

Knob 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 or 206-364-3196
Internet: mabry@mabry.com
Web: www.mabry.com

Where did you get this copy of Knob?

_____

Name: _____

Ship to: _____

_____

_____

_____

Phone: _____

Fax: _____

E-Mail: _____

MC/VISA/AMEX: _____ exp. _____

P.O. # (if any): _____ Signature _____

qty ordered ____ REGISTRATION
$20.00 ($25.00 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
$45.00 ($50.00 international). Check or money order in U.S. currency drawn on a U.S. bank. Add $5.00 per order for shipping and handling.

Close Copy Print

Indicator Properties Example

In this example, the program shows what happens when you change the look of the knob's indicator. To try this example, paste the code into the Declarations section of a form that contains a horizontal scroll bar, a label, two command buttons, a common dialog control, and a knob. Press F5. Play with the scroll bar and the command buttons.

```
Sub Command1_Click ()
    Knob1.Indicator = 1 - Knob1.Indicator
End Sub

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

    Knob1.IndicatorColor = CMDialog1.Color
End Sub

Sub Form_Load ()
    Form1.BackColor = &HC0C0C0

    Command1.Caption = "Change Style"
    Command1.Top = 720
    Command1.Left = 240
    Command1.Width = 1800
    Command1.Height = 360

    Command2.Caption = "Change Color"
    Command2.Top = 1200
    Command2.Left = 240
    Command2.Width = 1800
    Command2.Height = 360

    Label1.BackColor = &HC0C0C0
    Label1.Top = 240
    Label1.Left = 2160
    Label1.Height = 255
    Label1.Width = 4000

    HScroll1.Top = 240
    HScroll1.Left = 240
    HScroll1.Width = 1800
    HScroll1.Min = 0
    HScroll1.Max = 20
    HScroll1.Value = 2

    Knob1.Top = 1680
    Knob1.Left = 240
    Knob1.Width = 1800
    Knob1.Height = 1800
    Knob1.Radius = 600
End Sub

Sub HScroll1_Change ()
```

```
    Call HScroll1_Scroll
End Sub

Sub HScroll1_Scroll ()
    Knob1.IndicatorWidth = HScroll1.Value
    Label1.Caption = "IndicatorWidth: " & HScroll1.Value
End Sub
```

Close Copy Print

Knob Style Properties Example

In this example, the program shows what happens when you change the knob. To try this example, paste the code into the Declarations section of a form that contains two command buttons, a common dialog control, and a knob. Press F5. Play the command buttons.

```
Sub Command1_Click ()
    Knob1.KnobStyle = (Knob1.KnobStyle + 1) Mod 4
End Sub

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

    Knob1.KnobColor = CMDialog1.Color
End Sub

Sub Form_Load ()
    Form1.BackColor = &HC0C0C0

    Command1.Caption = "Change Style"
    Command1.Top = 720
    Command1.Left = 240
    Command1.Width = 1800
    Command1.Height = 360

    Command2.Caption = "Change Color"
    Command2.Top = 1200
    Command2.Left = 240
    Command2.Width = 1800
    Command2.Height = 360

    Knob1.Top = 1680
    Knob1.Left = 240
    Knob1.Width = 1800
    Knob1.Height = 1800
    Knob1.Radius = 600
End Sub
```

Close Copy Print

Radius Property Example

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

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

    HScroll1.Min = 100
    HScroll1.Max = 950
    HScroll1.Value = 200

    Knob1.Width = 2000
    Knob1.Height = 2000
    Knob1.Radius = HScroll1.Value

    Label1.Caption = Knob1.Radius
    Label1.BackColor = &HC0C0C0
End Sub

Sub HScroll1_Scroll ()
    Knob1.Radius = HScroll1.Value
    Label1.Caption = HScroll1.Value
End Sub
```

See Also

BorderWidth Property

KnobStyle Property

See Also

BevelWidth Property
Indicator Property

See Also

- **LinkControl** Property
- **LinkProperty** Property
- **Scroll** Event
- **Value** Property

See Also

BorderWidth Property
IndicatorColor Property
IndicatorWidth Property
Value Property

See Also

Indicator Property

IndicatorWidth Property

See Also

BorderWidth Property
Indicator Property
IndicatorWidth Property

See Also

IndicatorColor Property

KnobStyle Property

See Also

KnobColor Property

See Also

Value Property

See Also

Change Event

Value Property

See Also

TickCaptionColor Property
TickColor Property
TickCount Property
TickGap Property
TickLength Property
TickWidth Property

See Also

- **TickCaption** Property
- **TickColor** Property
- **TickCount** Property
- **TickGap** Property
- **TickLength** Property
- **TickWidth** Property

See Also

- **TickCaption** Property
- **TickCaptionColor** Property
- **TickCount** Property
- **TickGap** Property
- **TickLength** Property
- **TickWidth** Property

See Also

- **TickCaption** Property
- **TickCaptionColor** Property
- **TickColor** Property
- **TickGap** Property
- **TickLength** Property
- **TickWidth** Property

See Also

- **TickCaption** Property
- **TickCaptionColor** Property
- **TickColor** Property
- **TickCount** Property
- **TickLength** Property
- **TickWidth** Property

See Also

- **TickCaption** Property
- **TickCaptionColor** Property
- **TickColor** Property
- **TickCount** Property
- **TickGap** Property
- **TickWidth** Property

See Also

- **TickCaption** Property
- **TickCaptionColor** Property
- **TickColor** Property
- **TickCount** Property
- **TickGap** Property
- **TickLength** Property

See Also

- **Change** Event
- **LinkControl** Property
- **LinkProperty** Property
- **Max** Property
- **Min** Property
- **Scroll** Event

BevelWidth Property

See Also

Description

Determines the width of the knob's bevel.

Syntax

*object.***BevelWidth** [= *integer*]

The syntax of the **BevelWidth** property has these parts:

Part	Description
object	A Knob control.
integer	An integer that specifies the bevel's width, in pixels.

Remarks

This property determines the width of the bevel that surrounds the knob.

Data Type

Integer

BorderWidth Property

See Also

Description

Determines the distance between the inside of the bevel and the indicator.

Syntax

*object.***BorderWidth** [= *integer*]

The syntax of the **BorderWidth** property has these parts:

Part	Description
object	A Knob control.
integer	An integer that specifies the distance between the outer bevel and the indicator, in pixels.

Remarks

This property determines the distance between the bevel on the knob, and the outside edge of the indicator.

Data Type

Integer

Change Event

See Also

Description

Occurs when the value has changed.

Syntax

Sub *object*_**Change(**[*index* **As Integer**]**)**

The syntax of the **Change** event has these parts:

Part	Description
object	A Knob control.
index	An integer that identifies a control if it's in a control array.

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.

Indicator Property

See Also Indicator Properties Example

Description

Determines the style of indicator to use for the knob.

Syntax

*object.***Indicator** [= *integer*]

The syntax of the **Indicator** property has these parts:

Part	Description
object	A Knob control.
integer	An integer that determines the indicator style.

Remarks

The value of this property determines the style of indicator to use for the knob. The following styles are supported:

Constant	Value	Description
icSpot	0	Spot
icLine	1	Line

Data Type

Integer (enumerated)

IndicatorColor Property

See Also Indicator Properties Example

Description

Determines the color of the indicator.

Syntax

*object.***IndicatorColor** [= *color*]

The syntax of the **IndicatorColor** property has these parts:

Part	Description
object	A Knob control.
color	A color expression that determines the color of the indicator.

Remarks

This property determines the color of the indicator on the knob.

Data Type

Color

IndicatorWidth Property

See Also Indicator Properties Example

Description

Determines the width of the indicator.

Syntax

*object.***IndicatorWidth** [= *integer*]

The syntax of the **IndicatorWidth** property has these parts:

Part	Description
object	A Knob control.
integer	An integer that specifies the width fo the indicator.

Remarks

This property determines the width of the indicator on the knob. IndicatorWidth is measured in pixels.

Data Type

Integer

KnobColor Property

See Also Knob Style Properties Example

Description

Determines the color of the knob's face.

Syntax

*object.***KnobColor** [= *color*]

The syntax of the **KnobColor** property has these parts:

Part	Description
object	A Knob control.
color	A color expression that specifies the knob's face color.

Remarks

This property determines the color of the knob's face.

Data Type

Color

KnobStyle Property

See Also Knob Style Properties Example

Description

Determines the knob's visual style.

Syntax

*object.***KnobStyle** [= *integer*]

The syntax of the **KnobStyle** property has these parts:

Part	Description
object	A Knob control.
integer	An integer that determines the knob's style.

Remarks

This property determines the style of the knob. Valid values are:

Constant	Value	Description
kscNone	0	Normal frame
kscRaised	1	Raised
kscLowered	2	Lowered
kscTextured	3	Textured

Data Type

Integer

LinkControl and LinkProperty Properties

Description

Sets up link to another control.

Syntax

*object.***LinkControl**
*object.***LinkProperty**

The syntax of the **LinkControl** and **LinkProperty** properties has these parts:

Part	Description
object	A Knob control.

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.

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

Max and Min Properties

See Also

Description

Determines the range of values for this control.

Syntax

*object.***Max** [= *integer*]
*object.***Min** [= *integer*]

The syntax of the **Max** and **Min** properties has these parts:

Part	Description
object	A Knob control.
integer	An integer that specifies the lower or upper bound for this control's value.

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

Radius Property

Radius Property Example

Description

Determines the size of the knob.

Syntax

*object.***Radius** [= *radius*]

The syntax of the **Radius** property has these parts:

Part	Description
object	A Knob control.
radius	An expression that specifies the radius of the knob, in twips.

Remarks

This property determines the size of the knob.

When the knob is sized at design-time, this property is automatically scaled. (VBX only)

Data Type

Real

Scroll Event

See Also

Description

Occurs when the user changes the value.

Syntax

Sub *object*_**Scroll(**[*index* **As Integer**]**)**

The syntax of the **Scroll** event has these parts:

Part	Description
object	A Knob control.
index	An integer that identifies a control if it's in a control array.

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.

TickCaption Property

See Also

Description

Determines the captions on the tick marks.

Syntax

*object.***TickCaption(** *tickindex* **)** [= *string*]

The syntax of the **TickCaption** property has these parts:

Part	Description
object	A Knob control.
tickindex	An integer that identifies a specific tick entry.
string	A string expression that specifies the tick caption.

Remarks

This property array specifies the text that's associated with each tick mark. *tickIndex* is numbered from 0 to (TickCount - 1), starting at the left-bottom of the knob and moving around clock-wise.

You can set this property at design-time by selecting this property, and then pressing the ellipsis button. The dialog box that pops up lets you enter and edit captions. (VBX only)

Data Type

String

TickCaptionColor Property

See Also

Description

Determines the color of the tick caption text.

Syntax

*object.***TickCaptionColor** [= *color*]

The syntax of the **TickCaptionColor** property has these parts:

Part	Description
object	A Knob control.
color	A color expression that specifies the color of the tick captions.

Remarks

This property sets the color of the tick captions.

Data Type

Color

TickColor Property

See Also

Description

Determines the color of the tick marks.

Syntax

*object.***TickColor** [= *color*]

The syntax of the **TickColor** property has these parts:

Part	Description
object	A Knob control.
color	A color expression that specifies the color of the tick marks.

Remarks

This property specifies the color of the tick marks.

Data Type

Color

TickCount Property

See Also

Description

Specifies the number of tick marks.

Syntax

*object.***TickCount** [= *tickcount*]

The syntax of the **TickCount** property has these parts:

Part	Description
object	A Knob control.
tickcount	An integer that specifies the number of ticks marks.

Remarks

This property determines the number of tick marks and their associated captions.

Data Type

Integer

TickGap Property

See Also

Description

Determines the distance between the tick marks and the knob.

Syntax

*object.***TickGap** [= *integer*]

The syntax of the **TickGap** property has these parts:

Part	Description
object	A Knob control.
integer	An integer that specifies the space between the tick marks and the knob, in pixels.

Remarks

This property specifies the distance between the inside edge of the tick marks and the outside edge of the knob. This property is measured in pixels.

Data Type

Integer

TickWidth Property

See Also

Description

Determines the width of the tick marks.

Syntax

*object.***TickWidth** [= *integer*]

The syntax of the **TickWidth** property has these parts:

Part	Description
object	A Knob control.
integer	An integer that specifies the width of the tick marks, in pixels.

Remarks

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

Data Type

Integer

TickLength Property

See Also

Description

Determines the length of the tick marks.

Syntax

*object.***TickLength** [= *integer*]

The syntax of the **TickLength** property has these parts:

Part	Description
object	A Knob control.
integer	An integer that specifies the length of the tick marks, in pixels.

Remarks

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

Data Type

Integer

Value Property

See Also

Description

Specifies the current position of the indicator.

Syntax

*object.***Value** [= *integer*]

The syntax of the **Value** property has these parts:

Part	Description
object	A Knob control.
integer	An integer that specifies the current position of the indicator.

Remarks

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

This property must range from Max to Min.

Data Type

Integer

Version Property

Description

Shows the version of the control.

Syntax

*object.***Version**

The syntax of the **Version** property has these parts:

Part	Description
object	A Knob control.

Remarks

This property holds the current version of the control. It is read-only and available at both design-time and run-time.

Data Type

String

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 or 206-364-3196

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.

The sample versions of our products are intended for evaluation purposes only. You may not use the sample version to develop completed applications.