



Help for Indic

[Properties](#)

[Events](#)

[Frequently Asked Questions](#)

Registration Information

Order Form

Getting Custom Controls Written

Licensing Information

Description

Horizontal and vertical Indicator VBX/OCX custom controls allow you to create professional VU segmented meters. This custom control was specially designed to allow you to easily create state-of-the-art graphic LED VU meters. Three separate colored segment sections can be defined with up to twenty LED segments per section. A comprehensive set of 3D bevel properties allows for flexible visual control.

- * Horizontal and vertical indicator (otherwise known as a VU meter).
- * Display information (such as volume, balance, etc.)
- * A comprehensive set of 3D bevel properties allows for flexible visual control.
- * Three separate colored segment sections can be defined with up to twenty LED segments per section.

A Visual Basic sample project, VU Meter, is included to show off some of the properties of VINDIC. It allows you to play with the VU Indicator properties to see how cool you can make a VU meter look.

File Name

HINDIC.VBX, VINDIC.VBX, HINDIC16.OCX, VINDIC16.OCX, HINDIC32.OCX, VINDIC32.OCX

ActiveX / OCX Object Name

Mabry.HIndicCtrl / Mabry.VIndicCtrl

ActiveX Compatibility

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

HIndicator and VIndicator

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

Indic Properties

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

BackColor Property
*BevelInner Property
*BevelOuter Property
*BevelWidth Property
*Border Property
*BorderWidth Property
Enabled Property
Height Property
hWnd Property
Index Property
*ItemBackColor Property
*ItemCount1 Property
*ItemCount2 Property
*ItemCount3 Property
*ItemForeColor1 Property
*ItemForeColor2 Property
*ItemForeColor3 Property
Left Property
*LinkControl Property
*LinkProperty Property
*Max Property
*Min Property
Name Property
Parent Property
Tag Property
*ThreeD Property
Top Property
*Value Property
*Version Property
Visible Property
Width Property

Close

Indic Events

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

Click Event

DbClick 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

Indic 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 Indic (control only) is US\$15 (US\$20 for International orders). The cost of Indic and the C/C++ source code (of the control itself) is US\$40 (US\$45 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. Indic's SWREG ID number is 10289. The source code version's ID number is 10291. 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.

Copyright © 1996-1997 by Mabry Software, Inc.



Indic 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 Indic?

Name: _____

Ship to: _____

Phone: _____

Fax: _____

E-Mail: _____

MC/VISA/AMEX: _____ exp. _____

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

qty ordered _____ REGISTRATION
\$15.00 (\$20.00 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
\$40.00 (\$45.00 international). Check or money order in U.S. currency drawn on a U.S. bank. Add \$5.00 per order for shipping and handling.



Bevel Properties Example

In this example, the program shows what happens when you vary the bevels on the controls. To try this example, paste the code into the Declarations section of a form that has a horizontal indicator and a horizontal scroll bar. Press F5. Play with the scroll bar.

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

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

    HIndicator1.BackColor = &HC0C0C0
End Sub

Sub HScroll1_Scroll ()
    HIndicator1.BevelInner = HScroll1.Value
    HIndicator1.BevelOuter = HScroll1.Value
End Sub
```



ItemBackColor Property Example

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

```
Sub Command1_Click ()
    CMDialog1.Color = HIndicator1.ItemBackColor
    CMDialog1.Flags = 1
    CMDialog1.Action = 3

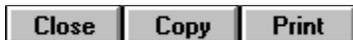
    HIndicator1.ItemBackColor = CMDialog1.Color
End Sub

Sub Form_Load ()
    Form1.BackColor = &HC0C0C0

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

    HIndicator1.Top = 720
    HIndicator1.Left = 240
    HIndicator1.Width = 3600
    HIndicator1.Height = 900
    HIndicator1.BevelInner = 3
    HIndicator1.BevelOuter = 1
    HIndicator1.BackColor = &HC0C0C0

    HIndicator1.ItemBackColor = &HC0C0C0
End Sub
```



ItemForeColor and ItmeCount Properties Example

In this example, the program shows what happens when you vary the color and count of the items in an indicator. To try this example, paste the code into the Declarations section of a form that contains three horizontal scroll bars, three labels, three command buttons, a common dialog box control, and a horizontal indicator control. Press F5. Play with the command buttons and the scroll bars.

```
Sub Command1_Click ()
    CMDialog1.Color = HIndicator1.ItemForeColor1
    CMDialog1.Flags = 1
    CMDialog1.Action = 3

    HIndicator1.ItemForeColor1 = CMDialog1.Color
End Sub
```

```
Sub Command2_Click ()
    CMDialog1.Color = HIndicator1.ItemForeColor2
    CMDialog1.Flags = 1
    CMDialog1.Action = 3

    HIndicator1.ItemForeColor2 = CMDialog1.Color
End Sub
```

```
Sub Command3_Click ()
    CMDialog1.Color = HIndicator1.ItemForeColor3
    CMDialog1.Flags = 1
    CMDialog1.Action = 3

    HIndicator1.ItemForeColor3 = CMDialog1.Color
End Sub
```

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

    HIndicator1.Top = 1680
    HIndicator1.Left = 240
    HIndicator1.Width = 6000
    HIndicator1.Height = 600
    HIndicator1.BevelInner = 3
    HIndicator1.BevelOuter = 1
    HIndicator1.Value = 100
    HIndicator1.BackColor = &HC0C0C0
    HIndicator1.ItemBackColor = &HC0C0C0
```

```
Command1.Top = 240
Command1.Left = 240
Command1.Width = 1800
Command1.Height = 360
```

```
Command1.Caption = "Change Color 1"
```

```
HScroll11.Top = 240
HScroll11.Left = 2160
HScroll11.Width = 900
HScroll11.Min = 0
```

```

HScroll11.Max = 20
HScroll11.Value = HIndicator1.ItemCount1

Label1.Top = 240
Label1.Left = 3180
Label1.Width = 2000
Label1.BackColor = &HC0C0C0

Command2.Top = 720
Command2.Left = 240
Command2.Width = 1800
Command2.Height = 360
Command2.Caption = "Change Color 2"

HScroll12.Top = 720
HScroll12.Left = 2160
HScroll12.Width = 900
HScroll12.Min = 0
HScroll12.Max = 20
HScroll12.Value = HIndicator1.ItemCount2

Label2.Top = 720
Label2.Left = 3180
Label2.Width = 2000
Label2.BackColor = &HC0C0C0

Command3.Top = 1200
Command3.Left = 240
Command3.Width = 1800
Command3.Height = 360
Command3.Caption = "Change Color 3"

HScroll13.Top = 1200
HScroll13.Left = 2160
HScroll13.Width = 900
HScroll13.Min = 0
HScroll13.Max = 20
HScroll13.Value = HIndicator1.ItemCount3

Label3.Top = 1200
Label3.Left = 3180
Label3.Width = 2000
Label3.BackColor = &HC0C0C0
End Sub

Sub HScroll11_Change ()
HIndicator1.ItemCount1 = HScroll11.Value
Label1.Caption = "ItemCount1: " & HScroll11.Value
End Sub
Sub HScroll11_Scroll ()
Call HScroll11_Change
End Sub

Sub HScroll12_Change ()
HIndicator1.ItemCount2 = HScroll12.Value
Label2.Caption = "ItemCount2: " & HScroll12.Value
End Sub

```

```
Sub HScroll12_Scroll ()  
    Call HScroll12_Change  
End Sub
```

```
Sub HScroll13_Change ()  
    HIndicator1.ItemCount3 = HScroll13.Value  
    Label3.Caption = "ItemCount3: " & HScroll13.Value  
End Sub
```

```
Sub HScroll13_Scroll ()  
    Call HScroll13_Change  
End Sub
```

See Also

[BevelOuter Property](#)

[BevelWidth Property](#)

[BorderWidth Property](#)

See Also

BevelInner Property

BevelWidth Property

BorderWidth Property

See Also

BevelInner Property

BevelOuter Property

Border Property

BorderWidth Property

See Also

BevelInner Property

BevelOuter Property

BevelWidth Property

BorderWidth Property

See Also

BevelInner Property

BevelOuter Property

See Also

[ItemForeColor1 Property](#)

[ItemForeColor2 Property](#)

[ItemForeColor3 Property](#)

See Also

[ItemBackColor](#) Property

[ItemForeColor1](#) Property

[ItemForeColor2](#) Property

[ItemForeColor3](#) Property

[Max](#) Property

[Min](#) Property

[Value](#) Property

See Also

[ItemBackColor](#) Property

[ItemCount1](#) Property

[ItemCount2](#) Property

[ItemCount3](#) Property

[Max](#) Property

[Min](#) Property

[Value](#) Property

See Also

Value Property

See Also

BevelInner Property

BevelOuter Property

See Also

[LinkControl Property](#)

[LinkProperty Property](#)

[Max Property](#)

[Min Property](#)

BevelInner Property

[See Also](#)

[Bevel Properties Example](#)

Description

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

Syntax

object.BevelInner [= *bevel*]

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

Part	Description
<i>object</i>	An Indic control.
<i>bevel</i>	An integer that determines the style of bevel.

Remarks

The value of this property determines the style of the inner border. This property may be one of four values:

Constant	Value	Description
bcNone	0	Normal frame
bcRaised	1	Raised frame (3-D)
bcInset	2	Inset frame (3-D)
bcLowered	3	Lowered frame (3-D)

Data Type

Integer (enumerated)

BevelOuter Property

[See Also](#)

[Bevel Properties Example](#)

Description

Determines the 3-D style of the border (if any) surrounding the control.

Syntax

object.**BevelOuter** [= *bevel*]

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

Part	Description
<i>object</i>	An Indic control.
<i>bevel</i>	An integer that determines the style of bevel.

Remarks

The value of this property determines the style of the outer bevel. This property may be one of four values:

Constant	Value	Description
bcNone	0	Normal frame
bcRaised	1	Raised frame (3-D)
bcInset	2	Inset frame (3-D)
bcLowered	3	Lowered frame (3-D)

Data Type

Integer (enumerated)

BevelWidth Property

[See Also](#)

[Bevel Properties Example](#)

Description

Determines the width of the inner and outer bevels.

Syntax

object.**BevelWidth** [= *width*]

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

Part	Description
<i>object</i>	An Indic control.
<i>width</i>	An integer that determines the width of the inner and outer bevels (in pixels).

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

Border Property

[See Also](#)

[Bevel Properties Example](#)

Description

Determines if a border is used.

Syntax

object.**Border** [= *border*]

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

<u>Part</u>	<u>Description</u>
<i>object</i>	An Indic control.
<i>border</i>	An integer that determines if a border is present.

Remarks

The value of this property determines if a border is present. If this property is set to None, no border (inner bevel or outer bevel) is displayed. This property may be one of the following values:

<u>Constant</u>	<u>Value</u>	<u>Description</u>
bNone	0	None
bSingle	1	Single width

Data Type

Integer (enumerated)

BorderWidth Property

[See Also](#)

[Bevel Properties Example](#)

Description

Determines the distance between the inner border and the outer border.

Syntax

object.**BorderWidth** [= *width*]

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

Part	Description
<i>object</i>	An Indic control.
<i>width</i>	An integer that determines the distance between the inner and outer bevels (in pixels).

Remarks

The value of this property determines the distance between the inner border (if any, see the [BevelInner property](#)) and the outer border (if any, see the [BevelOuter property](#)). This property is always measured in pixels.

Data Type

Integer

ItemBackColor Property

[See Also](#)

[ItemBackColor Property Example](#)

Description

Determines the color of the background of the items.

Syntax

object.ItemBackColor [= *background*]

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

Part	Description
<i>object</i>	An Indic control.
<i>background</i>	A color expression that determines the color of an indicator "light" when "off".

Remarks

This property specifies the color of the item backgrounds. The items are filled with this color when not "on" (i.e. filled with one of the [ItemForeColor](#)s).

Data Type

Color

ItemCount1, ItemCount2 and ItemCount3 Properties

See Also [ItemForeColor and ItmeCount Properties Example](#)

Description

Determines the number of items in the indicator.

Syntax

object.**ItemCount1** [= *count*]

object.**ItemCount2** [= *count*]

object.**ItemCount3** [= *count*]

The syntax of the **ItemCount1**, **ItemCount2** and **ItemCount3** properties has these parts:

Part	Description
<i>object</i>	An Indic control.
<i>count</i>	An integer that determines the number of items in each segment of the indicator.

Remarks

This property specifies the number of the items in the control. These properties must be greater than or equal to zero. If all three are zero, no items are displayed.

The first **ItemCount1** items are painted with [ItemForeColor1](#). The next **ItemCount2** items are painted with [ItemForeColor2](#). And, the remaining **ItemCount3** items are painted with [ItemForeColor3](#).

Data Type

Integer

ItemForeColor1, ItemForeColor2 and ItemForeColor3 Properties

See Also

[ItemForeColor and ItemCount Properties Example](#)

Description

Determines the color of the selected items.

Syntax

object.ItemForeColor1 [= *item*]

object.ItemForeColor2 [= *item*]

object.ItemForeColor3 [= *item*]

The syntax of the **ItemForeColor1**, **ItemForeColor2** and **ItemForeColor3** properties has these parts:

Part	Description
<i>object</i>	An Indic control.
<i>item</i>	A color expression that determines the color of the segments in the indicator.

Remarks

This property specifies the color of the items when they are selected (this is dependent upon the [Min](#), [Max](#), [Value](#), and [ItemCount](#) properties).

The first [ItemCount1](#) items are painted with ItemForeColor1. The next [ItemCount2](#) items are painted with ItemForeColor2. And, the remaining [ItemCount3](#) items are painted with ItemForeColor3.

Data Type

Color

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
<i>object</i>	An Indic 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** [= *value*]

object.**Min** [= *value*]

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

Part	Description
<i>object</i>	An Indic control.
<i>value</i>	An integer that determines the outside range of the 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

ThreeD Property

[See Also](#)

Description

Determines whether or not 3-D styles are used.

Syntax

object.**ThreeD** [= *threed*]

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

Part	Description
<i>object</i>	An Indic control.
<i>threed</i>	A boolean expression that determines if 3-D styles are used.

Remarks

If this property is set to False, no 3-D style bevels are used. If this property is set to True, any bevel can be used.

This property is only present in the VBX versions.

Data Type

Integer (boolean)

Value Property

[See Also](#)

Description

Specifies the current position of the indicator.

Syntax

object.**Value** [= *value*]

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

Part	Description
<i>object</i>	An Indic control.
<i>value</i>	An integer that determines the current position of the control.

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:

<u>Part</u>	<u>Description</u>
<i>object</i>	An Indic 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.

