



## Help for TrueBar VBX

[Properties](#)   [Events](#)

### Description

TrueBar VBX makes bar code generation easy. Just pick the orientation, set the size, and pick the bar code style. Then, set the caption to whatever you want the bars to be.

TrueBar VBX uses the Caption property to determine what to display. ForeColor is used for the color of the Bars. BackColor is used for the color of the background. If TrueBar VBX cannot fit the entire bar code in the control window, it will fill the entire control with the background color.

Under VB 3.0, TrueBar VBX will bind to a data control. It uses the DataField, DataSource, and DataChanged properties.

### File Name

TRUEBAR.VBX

### Object Type

Barcode

### Compatibility

Level 1 and up

## Properties

All of the properties that apply to this control are in this table. Properties that have special meaning for this control or that only apply to this control are marked with an asterisk (\*).

BackColor	HelpContextID	<u>*PrinterScaleMode</u>
<u>*BarWidth</u>	Height	<u>*PrinterTop</u>
Caption	hWnd	<u>*PrinterWidth</u>
DataChanged	Index	<u>*Style</u>
DataField	Left	Tag
DataSource	Name	Top
<u>*Direction</u>	Parent	Visible
<u>*Displayed</u>	<u>*PrinterHDC</u>	<u>*UPCNotches</u>
Enabled	<u>*PrinterHeight</u>	Width
ForeColor	<u>*PrinterLeft</u>	

Caption is the default value for the control.

BarWidth Property

[See Also](#)      [Example](#)

**Description**

Determines the width of the bars in the bar code.

**Usage**

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

**Remarks**

This represents the width of the "narrow" bar in the code selected (see [Style](#)). If this is set to zero (0), TrueBar VBX expands the bar code to fit in the control.

**Data Type**

Integer

See Also

Properties:

[Direction](#)

[Style](#)



## BarWidth Property Example

In this example, the program will vary the width of the bars. To try this example, paste the code into the Declarations section of a form that contains a horizontal scroll bar, a label (which shows the current BarWidth property), and a TrueBar VBX control (make this control wide). Press F5. Play with the scroll bar to see things change.

```
Sub Form_Load ()
    HScroll11.Value = 0
    HScroll11.Min = 0
    HScroll11.Max = 10

    Label1.Caption = "0"

    Barcode1.Style = 3
    Barcode1.BarWidth = 0
    Barcode1.Caption = "SAMPLE"
End Sub

Sub HScroll11_Change ()
    Barcode1.BarWidth = HScroll11.Value
    Label1.Caption = Format$( HScroll11.Value )
End Sub

Sub HScroll11_Scroll ()
    Call HScroll_Change
End Sub
```

Direction Property

[See Also](#)      [Example](#)

### **Description**

Determines the direction/orientation of the bar code.

### **Usage**

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

### **Remarks**

The value of this property determines what direction the bar code goes. This property can be one of following values:

<b>Value</b>	<b>Description</b>
0	Left-to-right
1	Right-to-left
2	Top-to-bottom
3	Bottom-to-top

### **Data Type**

Integer (enumerated)

See Also

Properties:

BarWidth

Style



## Direction Property Example

In this example, the program will vary the direction the bar code is painted/printed. To try this example, paste the code into the Declarations section of a form that contains a horizontal scroll bar, a label (which shows the current Direction property), and a TrueBar VBX control (make this control tall and wide). Press F5. Play with the scroll bar to see things change.

```
Sub Form_Load ()
    HScroll1.Value = 0
    HScroll1.Min = 0
    HScroll1.Max = 3

    Label1.Caption = "0"

    Barcode1.Style = 3
    Barcode1.BarWidth = 2
    Barcode1.Direction = 0
    Barcode1.Caption = "SAMPLE"
End Sub

Sub HScroll1_Change ()
    Barcode1.Direction = HScroll1.Value
    Label1.Caption = Format$( HScroll1.Value )
End Sub

Sub HScroll1_Scroll ()
    Call HScroll_Change
End Sub
```



Displayed Property

[See Also](#)      [Example](#)

### **Description**

Returns the actual string displayed in the bar code.

### **Usage**

[*form.*][*control.*]**Displayed**

### **Remarks**

This property holds what was actually displayed. Bar codes don't always have all the characters available. Some are purely numeric, some have numbers and letters, etc. This property lets your program know what was actually displayed.

This property is read-only.

### **Data Type**

String

See Also  
Properties:  
Caption  
Style



## Displayed Property Example

In this example, the program will show what is displayed when various strings are attempted. To try this example, paste the code into the Declarations section of a form that contains a label (which shows the current Displayed property), a text box, and a TrueBar VBX control. Press F5. Play with the edit box and see what appears in the label.

```
Sub Form_Load ()
    Label1.Caption = "0"
    Text1.Text = "SAMPLE"

    Barcode1.Style = 3
    Barcode1.BarWidth = 2
    Barcode1.Direction = 0
    Barcode1.Caption = Text1.Text
End Sub

Sub Text1_Change ()
    Barcode1.Caption = Text1.Text
    Label1.Caption = Barcode1.Displayed
End Sub
```

PrinterHDC Property

[See Also](#)      [Example](#)

### **Description**

Causes the bar code to be painted in the device handle specified (usually the printer object).

### **Usage**

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

### **Remarks**

When this property is set, TrueBar VBX paints itself into the Windows device context specified. Normally, this is used to print the barcode. To do this, use the hDC property of VB's Printer.

To print a bar code, make sure the Caption property is set correctly. Then, set the PrinterHeight, PrinterLeft, PrinterTop, and PrinterWidth properties. Finally, set Barcode1.PrinterHDC to Printer.hDC.

### **Data Type**

Integer

See Also

Objects:

Printer Object

Properties:

hDC

PrinterLeft, PrinterTop, PrinterRight,

PrinterBottom

PrinterScaleMode

Style

PrinterLeft, PrinterTop, PrinterWidth, and PrinterHeight Properties  
[See Also](#)      [Example](#)

### **Description**

Describes the area in which to paint the barcode when printing.

### **Usage**

```
[form.][control.]PrinterLeft[ = x ]  
[form.][control.]PrinterTop[ = y ]  
[form.][control.]PrinterWidth[ = xx ]  
[form.][control.]PrinterHeight[ = ny ]
```

### **Remarks**

These properties describe where to place the barcode when printing. The coordinates used should be in the units specified by the [PrinterScaleMode](#) property.

To print a bar code, make sure the Caption property is set correctly. Then, set the PrinterHeight, PrinterLeft, PrinterTop, and PrinterWidth properties. Finally, set Barcode1.PrinterHDC to Printer.hDC.

### **Data Type**

Integer (long)

See Also

Properties:

Caption

PrinterHDC

PrinterScaleMode

Style

## PrinterScaleMode Property

[See Also](#)      [Example](#)

### Description

Determines the units of measurement used by the [PrinterLeft](#), [PrinterTop](#), [PrinterWidth](#), and [PrinterHeight](#) properties.

### Usage

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

### Remarks

This property determines what units to use for the properties describing the area to print in. This property can be one of:

<b>Value</b>	<b>Description</b>
0	Pixel (default, smallest unit of printer resolution)
1	Twip (1440 twips per inch)
2	Point (72 points per inch)
3	Pixel (smallest unit of printer resolution)
4	Character (12 per inch horizontally, 6 per inch vertically)
5	Inch
6	Millimeter
7	Centimeter

To print a bar code, make sure the [Caption](#) property is set correctly. Then, set the [PrinterHeight](#), [PrinterLeft](#), [PrinterTop](#), and [PrinterWidth](#) properties. Finally, set [Barcode1.PrinterHDC](#) to [Printer.hDC](#).

### Data Type

Integer (enumerated)



See Also

Properties:

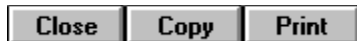
Caption

PrinterHDC

PrinterLeft, PrinterTop, PrinterWidth,

PrinterHeight

Style



## Printing Example

In this example, the program will print a bar code on your printer. The bar code and its style are determined by an edit box and a horizontal scroll bar. To try this example, paste the code into the Declarations section of a form that contains a text box, a command button, two labels, a horizontal scroll bar, and a TrueBar VBX control. Press F5. Then, set up the bar code using the horizontal scroll bar and the text box. Once you have a bar code you like, press the command button to print it.

```
Sub Form_Load ()
    HScroll1.Min = 0
    HScroll1.Max = 3
    HScroll1.Value = 3

    Text1.Text = "SAMPLE"

    Barcode1.Style = 3
    Barcode1.BarWidth = 0
    Barcode1.Direction = 0
    Barcode1.Caption = Text1.Text

    Label1.Caption = Barcode1.Displayed
    Label2.Caption = "3"
End Sub

Sub Command1_Click ()
    Screen.MousePointer = 11

    Printer.ScaleMode = 1
    Barcode1.PrinterScaleMode = Printer.ScaleMode

    Barcode1.PrinterLeft = 1440
    Barcode1.PrinterTop = 1440
    Barcode1.PrinterWidth = 3 * 1440
    Barcode1.PrinterHeight = 720
    Barcode1.PrinterHDC = Printer.hDC

    Printer.CurrentX = Barcode1.PrinterLeft
    Printer.CurrentY = Barcode1.PrinterTop + Barcode1.PrinterHeight +
        180
    Printer.Print Barcode1.Displayed

    Printer.NewPage
    Printer.EndDoc

    Screen.MousePointer = 0
End Sub

Sub Text1_Change ()
    Barcode1.Caption = Text1.Text
    Label1.Caption = Barcode1.Displayed
End Sub

Sub HScroll1_Change ()
    Barcode1.Style = HScroll1.Value
```

```
Label1.Caption = Barcode1.Displayed
Label2.Caption = Format$( HScroll1.Value )
End Sub

Sub HScroll1_Scroll ()
Call HScroll1_Change
End Sub
```

Style Property

[See Also](#)      [Example](#)

### Description

Determines what bar code style is used.

### Usage

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

### Remarks

This property determines the style, or symbology of the bar code used. Each style has a different set of characters that it supports. The styles are:

<b>Value</b>	<b>Description</b>
0	None
1	<u>2 of 5</u> - numbers only
2	<u>Interleaved 2 of 5</u> - numbers only
3	<u>3 of 9</u> - numbers, upper case, and - . * \$ / + % (default)
4	<u>Codabar (rationalized)</u> - numbers and A B C D - \$ : / . +
5	<u>Extended 3 of 9</u> - full ASCII character set
6	<u>Code 128A</u> - upper case, numbers, punctuation, and control characters
7	<u>Code 128B</u> - upper and lower case, numbers, and punctuation
8	<u>Code 128C</u> - numbers only
9	<u>UPC-A</u> - numbers only (11 numbers and a check digit)
10	<u>MSI (Plessey)</u> - numbers only
11	<u>Code 93</u> - numbers, upper case, and - . * \$ / + %
12	<u>Extended Code 93</u> - full ASCII character set
13	<u>EAN-13</u> - numbers only (12 numbers and a check digit)
14	<u>EAN-8</u> - numbers only (7 numbers and a check digit)

### Data Type

Integer (enumerated)

See Also

Properties:

BarWidth

Direction

UPCNotches

Code 2 of 5



Interleaved 2 of 5



Extended Code 3 of 9





Code 3 of 9



Extended Code 93



EAN-13



EAN-8



Code 93



Code 128A



Code 128B



Code 128C





UPC-A



MSI (Plessey)



Rationalized Codabar





## Style Property Example

In this example, the program will vary the style (or symbology) of the bar code used. To try this example, paste the code into the Declarations section of a form that contains a horizontal scroll bar, two labels (the first shows the Style property and the second shows the current Displayed property), a text box, and a TrueBar VBX control. Press F5. Play with the text box and the scroll bar and see what appears in the second label.

```
Sub Form_Load ()
    HScroll1.Min = 0
    HScroll1.Max = 10
    HScroll1.Value = 3

    Text1.Text = "SAMPLE"

    Barcode1.Style = 3
    Barcode1.BarWidth = 2
    Barcode1.Direction = 0
    Barcode1.Caption = Text1.Text

    Label1.Caption = Barcode1.Displayed
    Label2.Caption = "3"
End Sub

Sub Text1_Change ()
    Barcode1.Caption = Text1.Text
    Label1.Caption = Barcode1.Displayed
End Sub

Sub HScroll1_Change ()
    Barcode1.Style = HScroll1.Value
    Label1.Caption = Barcode1.Displayed
    Label2.Caption = Format$( HScroll1.Value )
End Sub

Sub HScroll1_Scroll ()
    Call HScroll1_Change
End Sub
```

UPCNotches Property  
[See Also](#)

### **Description**

Determines where the notches are on a UPC or EAN barcode.

### **Usage**

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

### **Remarks**

This property determines where the notches are (if any) on a UPC or EAN barcode. The settings are:

<b>Value</b>	<b>Description</b>
0	None
1	Above
2	Below
3	Both (default)

### **Data Type**

Integer (enumerated)

See Also  
Properties:  
Style

## Events

All of the standard events that apply to this control are in this table.

Click

DblClick

