

Contents

BarCode/VBX is a custom control that is used in Visual Basic and Visual C++ to display and print barcodes. Several types of barcodes are supported, some of which allow several variations. BarCode/VBX is a bound control allowing you to bind the control to a data field. You have control of how the bar code is displayed and printed through the use of various properties, events, and functions.

Properties

Events

Functions

ReadMe

Events

The following events are generated by BarCode/VBX. You may choose to respond to these events in your code to provide increased functionality.

Changed

Click

DbClick

DragDrop

DragOver

LinkClose

LinkError

LinkNotify

LinkOpen

MouseDown

MouseMove

MouseUp

Resized

Properties

The following properties are available for your use. By setting these properties, you have complete control over how the barcode is displayed and printed.

Action
AlignH
AlignV
AutoSize
BarCodeType
BorderStyle
Caption
CaptionPosition
Checksum
ChecksumString
IntercharacterGap
PrinterHDC
PrinterTop
PrinterLeft
PrinterWidth
Ratio
Rotation
ScaleBarHeight
ScaleHeight
ScaleLeft
ScaleMode
ScaleNarrowBarWidth
ScaleOffsetH
ScaleOffsetV
ScaleTop
ScaleWidth
Text

Functions

The following functions are available for use with BarCode/VBX. They are defined in the 'barcode.bas' file.

BarCodeGetLastErrorCode
BarCodeGetLastErrorString
BarCodeGetLastErrorStringC
BarCodePrint
BarCodeWidth

Changed

This event is fired when the text property is changed.

Resized

This event is fired when the size of the barcode is changed. BarCode/VBX will automatically resize the container (if the AutoSize property is set to TRUE) in order to display the entire barcode. This event can be useful if you need to reposition other objects on your form based on the size of the barcode.

Click

This event is fired when the user clicks the left mouse button. This is a standard event, see the Visual Basic users manual for more information.

DbClick

This event is fired when the user double clicks the left mouse button. This is a standard event, see the Visual Basic users manual for more information.

DragDrop

This event is fired when the control is the target of a drop operation. This is a standard event, see the Visual Basic users manual for more information.

DragOver

This event is fired when the control is the target of a drop operation. This is a standard event, see the Visual Basic users manual for more information.

LinkClose

This event is fired when a DDE conversation is terminated. This is a standard event, see the Visual Basic users manual for more information.

LinkOpen

This event is fired when a DDE conversation is initiated. This is a standard event, see the Visual Basic users manual for more information.

LinkError

This event is fired when a DDE error occurs. This is a standard event, see the Visual Basic users manual for more information.

LinkNotify

This event is fired when a DDE conversation has been initiated and the data in the server has changed. This is a standard event, see the Visual Basic users manual for more information.

MouseDown

Fired when a mouse button is pressed. The mouse is captured as a result of this event. This is a standard event, see the Visual Basic users manual for more information.

MouseMove

Fired when mouse movement occurs. This is a standard event, see the Visual Basic users manual for more information.

MouseUp

Fired when a mouse button is released. The mouse capture is released as a result of this event. This is a standard event, see the Visual Basic users manual for more information.

Action Property

Description

This property executes a desired action.

Usage

`[form.]control.Action[= numericexpression]`

Remarks

By setting this property to the appropriate value, you can perform certain actions as shown in the table below.

Setting	Description
1	Copies the barcode to clipboard as a bitmap.
2	Copies the barcode to clipboard as a metafile.

Data Type

Integer

AlignH Property

Description

This property controls the horizontal position of the barcode within the container window.

Usage

[form.]control.AlignH[= *numericexpression*]

Remarks

By setting this property, you can position the barcode within the container window. The AutoSize property must be set to 'False' for this property to have any effect. If you choose '3 - Offset', the horizontal offset is defined by the ScaleOffsetH property. This property is not used when printing the barcode.

Setting	Description
0	Left
1	Right
2	Center
3	Offset

Data Type

Integer

AlignV Property

Description

This property controls the vertical position of the barcode within the container window.

Usage

[form.]control.AlignV[= *numericexpression*]

Remarks

By setting this property, you can position the barcode within the container window. The AutoSize property must be set to 'False' for this property to have any effect. If you choose '3 - Offset', the vertical offset is defined by the ScaleOffsetV property. This property is not used when printing the barcode.

Setting	Description
0	Top
1	Bottom
2	Center
3	Offset

Data Type

Integer

AutoSize Property

Description

This property controls whether the container window is automatically resized to the size of the barcode.

Usage

[*form.*]control.AutoSize[= *numericexpression*]

Remarks

This property controls how the container window is resized when the barcode changes and whether it should display the barcode using the full height of the container window. When using the Caption property, set this property to 0 or 2 so the barcode is not displayed using the full height of the container window and allowing the caption to be visible.

If the property is zero, the container window is not resized at all. Use caution when setting this property, as the container window may be too small to show the entire barcode. This property is not used when printing the barcode.

Setting	Description
0	None (no resizing will take place)
1	Height (the barcode will be displayed using the full height of the container window)
2	Width (the container window will adjust it's width to the barcode width)
3	Both (the width will be adjusted and the barcode will be displayed the full height of the container window)

Data Type

Integer

BorderStyle Property

Description

This property controls whether the container window has a border.

Usage

`[form.]control.BorderStyle [= boolean]`

Remarks

This property allows you to specify if the container window has a border. This property is only used when in design mode and can be helpful by showing the size and position of the barcode windows on your form.

Data Type

Integer (Boolean)

Caption Property

Description

Sets the human readable text for the barcode.

Usage

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

Remarks

This property sets the human readable text that will print underneath the barcode. Make sure that you have set the [AutoSize](#), [AlignH](#), [AlignV](#) and [ScaleBarHeight](#) properties to values that will allow the caption to be visible.

If you have created a barcode that is rotated and have selected a font that Windows cannot rotate, Barcode/VBX will use the Arial True Type font. Since Windows is not able to rotate some fonts, this behavior is necessary. The best way to ensure that the font is able to be rotated properly is to select a True Type font.

Data Type

String

CaptionPosition Property

Description

Determines the position of the caption for the barcode.

Usage

[*form.*]control.CaptionPosition= *setting*]

Remarks

The CaptionPosition property allows you to specify the position of the caption in relation to the barcode. The Caption property must be set to a valid value.

Setting	Description
0	Bottom Left
1	Bottom Center
2	Bottom Right
3	Top Left
4	Top Center
5	Top Right

Data Type

Integer

ScaleBarHeight Property

Description

Sets the height of the barcode.

Usage

[*form.*]control.ScaleBarHeight[= *boolean*]

Remarks

This property sets the height of the barcode if the AutoSize property is False. The unit of measurement is defined with the ScaleMode property.

Data Type

Single

ScaleHeight Property

Description

Sets the height of the barcode container window.

Usage

[*form.*]control.ScaleHeight[= *numericexpression*]

Remarks

This property sets the height of the barcode container window. The unit of measurement is defined with the ScaleMode property.

Data Type

Single

ScaleLeft Property

Description

Shows the left position of the barcode container window.

Remarks

This property shows the left position of the barcode container window and is not used for any other purpose at this time. The unit of measurement is defined with the ScaleMode property.

ScaleMode Property

Description

Determines the unit of measurement for sizing and positioning the barcode.

Usage

[*form.*]control.ScaleMode[= *setting*]

Remarks

The ScaleMode property allows you to specify the unit of measurement that will be used when sizing and positioning the barcode on the printer.

Setting	Description
0	Pixel (smallest unit of printer resolution).
1	(Default) Twip (1440 twips per logical inch; 567 twips per logical centimeter).
2	Point (72 points per logical inch).
3	Pixel (smallest unit of printer resolution).
4	Character (horizontal = 120 twips per unit; vertical = 240 twips per unit).
5	Inch.
6	Millimeter.
7	Centimeter.

Data Type

Integer

ScaleNarrowBarWidth Property

Description

Sets the width of the narrow bar for the barcode.

Usage

[*form.*]control.ScaleNarrowBarWidth[= *numericexpression*]

Remarks

This property sets width of the narrow bar. The unit of measurement is defined with the ScaleMode property.

Data Type

Single

ScaleOffsetH Property

Description

Sets the horizontal offset of the barcode within the container window.

Usage

*[form.]control.ScaleOffsetH[= *boolean*]*

Remarks

This property sets the horizontal offset of the barcode when the AutoSize property is false the AlignH property is set to '3 - Offset'. The unit of measurement is defined with the ScaleMode property.

Data Type

Single

ScaleOffsetV Property

Description

Sets the vertical offset of the barcode within the container window.

Usage

*[form.]control.ScaleOffsetV[= *boolean*]*

Remarks

This property sets the vertical offset of the barcode when the AutoSize property is false the the AlignV property is set to '3 - Offset'. The unit of measurement is defined with the ScaleMode property.

Data Type

Single

ScaleTop Property

Description

Shows the top position of the barcode container window.

Remarks

This property shows the top position of the barcode container window and is not used for any other purpose at this time. The unit of measurement is defined with the ScaleMode property.

ScaleWidth Property

Description

Sets the width of the barcode container window.

Usage

`[form.]control.ScaleWidth[= numericexpression]`

Remarks

This property sets the width of the barcode container window. The unit of measurement is defined with the ScaleMode property.

Data Type

Single

BarCodeType Property

Description

Sets the type of barcode used.

Usage

[*form.*]control.BarCodeType[= *setting*]

Remarks

By setting this property, you select the type of barcode that is displayed or printed. the following is a list of the possible types of barcodes.

Setting	Description
0	<u>Code 3 of 9</u>
1	<u>Extended Code 3 of 9</u>
2	<u>Interleaved 2 of 5</u>
3	<u>Code 93</u>
4	<u>Extended Code 93</u>
5	<u>UPCA</u>
6	<u>UPCE 10 digit</u>
7	<u>UPCE0 6 digit</u>
8	<u>UPCE1 6 digit</u>
9	<u>EAN 13</u>
10	<u>EAN 8</u>
11	<u>Code 128 Auto</u>
12	<u>Code 128 A</u>
13	<u>Code 128 B</u>
14	<u>Code 128 C</u>
15	<u>Codabar</u>
16	<u>MSI Plessey</u>
17	<u>UCC-128</u>
18	<u>POSTNET (Zip + 4 PostalCode)</u>

Data Type

Integer

Checksum Property

Description

Controls how the checksum is created

Usage

`[form.]control.Checksum[= setting]`

Remarks

Checksums can be optionally added to some barcodes. See the description on the particular bar code type for more information.

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

0	(Default) No checksum.
1	See the appropriate barcode type for more information.
2	See the appropriate barcode type for more information.

Data Type

Integer

Rotation Property

Description

Controls the rotation of the barcode.

Usage

`[form.]control.Rotation[= setting]`

Remarks

The bar code can be rotated by setting this property to the proper value. The following list shows the different orientations.

Setting	Description
0	Normal
1	Right
2	Upside Down
3	Left

Normal



Upside Down

Data Type

Integer

Text Property

Description

Sets the text to be used in creating the bar code.

Usage

`[form.]control.Text[= stringexpression]`

Remarks

The text property allows you to set the text that will be used to generate the bar code itself. If you are in design mode, changing this property will cause the BarCode/VBX to draw the barcode, if the text entered is valid for the bar code type that is currently selected.

Data Type

String

Ratio Property

Description

Sets the ratio of the barcode.

Usage

`[form.]control.Ratio[= setting]`

Remarks

The ratio of the wide bars to narrow bars can be controlled using this property. BarCode/VBX defaults to a 3:1 ratio. Valid selections for this property are listed below. This property only affects Code 3 of 9, Extended Code 3 of 9, and Interleaved 2 of 5.

Setting	Description
0	3:1
1	2.5:1
2	2:1

Data Type

Integer

ChecksumString Property

Description

Contains the checksum string

Usage

```
[stringexpression = ][form.]control.ChecksumString
```

Remarks

This property is a string that contains the value of the checksum character. This property is read-only at run time and not available at design time.

Data Type

String

PrinterTop Property

Description

Sets the top position of the barcode when printing.

Usage

[*form.*]control.PrinterTop[= *numericexpression*]

Remarks

The PrinterTop, PrinterLeft and PrinterHDC properties are used to print a barcode on the printer. The PrinterTop property allows you to specify the top position of the bar code. The unit of measure for this property is set using the ScaleMode property.

See Also:

PrinterWidth

Data Type

Single

PrinterLeft Property

Description

Sets the left position of the barcode when printing.

Usage

[*form.*]control.PrinterLeft[= *numericexpression*]

Remarks

The PrinterTop, PrinterLeft, and PrinterHDC properties are used to print a barcode on the printer. The PrinterLeft property allows you to specify the left position of the bar code. The unit of measure for this property is set using the ScaleMode property.

See Also:

PrinterWidth

Data Type

Single

PrinterWidth

Description

Retrieves the width of the barcode, in pixels.

Usage

numericexpression = [*form.*]*control*.PrinterWidth

Remarks

The PrinterWidth property will allow you to retrieve the width of the barcode after it has been printed by setting the PrinterHDC property. This value will be the width of the barcode, in pixels, as printed on the printer.

Example:

```
Dim theWidthInPixels%, theWidthInTwips%

Barcode1.Text = "123456"
Barcode1.PrinterLeft = 200
Barcode1.PrinterTop = 400
Barcode1.PrinterHDC = Printer.hDC

theWidthInPixels = Barcode1.PrinterWidth

theWidthInTwips = theWidthInPixels * Printer.TwipsPerPixelX

Printer.EndDoc
```

See Also:

[PrinterTop](#)
[PrinterLeft](#)
[PrinterHDC](#)

Data Type

Integer

PrinterHDC Property

Description

Sets the handle to the printer device context.

Usage

```
[form.]control.PrinterHDC[= hDC]
```

Remarks

The PrinterTop, PrinterLeft, and PrinterHDC properties are used to print a barcode on the printer. The PrinterHDC property allows you to specify the handle of the device context for the printer. When this property is set, the barcode will be printed on the printer. Note that the unit of measure for PrinterTop and PrinterLeft properties are defined by the ScaleMode property.

See Also:

PrinterWidth

Data Type

hDC

Example:

```
BarCode1.PrinterTop = 100  
BarCode1.PrinterLeft = 0  
BarCode1.PrinterHDC = Printer.hDC  
If BarCodeGetLastErrorCode() <= 0 Then  
    MsgBox BarCodeGetLastErrorString()  
End If
```

IntercharacterGap

Description

Sets the intercharacter gap on Code 3 of 9 and Codabar barcodes.

Usage

`[form.]control.IntercharacterGap[= setting]`

Remarks

By setting this property, you can control the intercharacter gap. This is the gap between the digits of the barcode.

Data Type

Integer

Code3 of 9



This bar code is an alphanumeric bar code allowing uppercase letters and numbers. BarCode/VBX will convert any lower case letters into upper case before printing the bar code. Each character consists of nine elements. 3 of the nine elements are wide, hence the name '3 of 9'.

Set the checksum property to a non-zero value to add a checksum to the barcode.

Extended 3 of 9



Extended 3 of 9 is similar to Code 3 of 9 except that it allows the full 128 ASCII character set to be encoded by printing two bar code characters for each text character.

Set the checksum property to a non-zero value to add a checksum to the barcode.

Interleaved 2 of 5



This is strictly a numeric bar code. Each encoded character is made up of five elements, two are wide and three are narrow. The number of characters to be printed must be an even number. If the number of characters to be printed is odd a zero will be appended to the beginning of the code.

Set the checksum property to a non-zero value to add a checksum to the barcode.

Code 93



Code 93 is an alpha-numeric bar code allowing upper case letters and numbers. BarCode/VBX will convert lower case letters to upper case before encoding them.

Set the checksum property to a non-zero value to add a checksum to the barcode.

Extended Code 93



Extended Code 93 is similar to Code 93 except that it allows the full 128 character ASCII character set to be encoded.

Set the checksum property to a non-zero value to add a checksum to the barcode.

UPCA



UPC (Universal Product Code) version A is used to encode an 11 digit number. The first digit is the system number and the rest are data characters. Both 2 and 5 digit supplementals are also supported.

The value of the checksum property is not used.

UPCE 10 digit



UPCE is a zero suppressed version of the UPCA barcode. This version allows 10 digits to be encoded. The first digit must be zero. Both 2 and 5 digit supplementals are also supported.

The value of the checksum property is not used.

UPCE0 6 digit



UPCE is a zero suppressed version of the UPCA barcode. This version allows 6 digits to be encoded. The first digit must be zero. Both 2 and 5 digit supplementals are also supported.

The value of the checksum property is not used.

UPCE1 6 digit



UPCE is a zero suppressed version of the UPCA barcode. This version allows 6 digits to be encoded. The first digit must be zero. Both 2 and 5 digit supplementals are also supported.

The value of the checksum property is not used.

EAN 13



EAN barcodes are used when the country origin needs to be known. There are 13 digits in EAN 13 where the first two characters are used to define the country of origin, the next ten are data, followed by the checksum. Both 2 and 5 digit supplementals are also supported.

The value of the checksum property is not used.

EAN 8



EAN barcodes are used when the country origin needs to be known. There are 8 digits in EAN 8 where the first two characters are used to define the country of origin, the next 5 are data, followed by the checksum. Both 2 and 5 digit supplementals are also supported.

The value of the checksum property is not used.

Code 128 Auto



Code 128 is a variable length bar code that is capable of encoding the entire 128 character ASCII character set. Code 128 allows three subsets, A, B and C. This version, 'Code 128 Auto', will automatically select the subset that will produce the smallest bar code.

Set the checksum property to a non-zero value to add a checksum to the barcode.

Code 128 A



Code 128 is a variable length bar code that is capable of encoding the entire 128 character ASCII character set. Code 128 allows three subsets, A, B and C. This subset (A) allows all standard upper case alpha-numeric keyboard characters plus control characters.

Set the checksum property to a non-zero value to add a checksum to the barcode.

Code 128 B



Code 128 is a variable length bar code that is capable of encoding the entire 128 character ASCII character set. Code 128 allows three subsets, A, B and C. This subset (B) allows all standard upper case alpha-numeric keyboard characters and lower case alpha characters.

Set the checksum property to a non-zero value to add a checksum to the barcode.

Code 128 C



Code 128 is a variable length bar code that is capable of encoding the entire 128 character ASCII character set. Code 128 allows three subsets, A, B and C. This subset (C) includes a set of 100 digit pairs from 00 to 99 inclusive. This allows double density numeric digits, two digits per bar coded character.

Set the checksum property to a non-zero value to add a checksum to the barcode.

Codabar



Codabar is a variable length barcode that can encode 16 data characters including 0-9, plus the symbols '-' '\$' ; '/' . '+'. There are 4 Start/Stop characters which are 'A' through 'D'. Codabar is used primarily for numeric data.

Set the checksum property to a non-zero value to add a checksum to the barcode.

MSI Plessey



This barcode is a variable length barcode that can encode up to 15 numeric digits. Checksum generation is dependent on the value of the checksum parameter. The following table indicates the value of the checksum property and the type of checksum created.

Setting	Description
0	one modulus 10 checksum
1	two modulus 10 checksums
2	one modulus 11 checksum/one modulus 10 checksum

UCC-128



This bar code is a specially defined subset of Code 128 that is used mostly on shipping containers. It is numeric only having a fixed length of 19 digits.

Set the checksum property to a non-zero value to add a checksum to the barcode.

POSTNET (Zip + 4 Postal Code)



The POSTNET barcode is used on envelopes and postcards that are sent through the U.S. Postal Service. This barcode is placed in the lower right hand corner of the envelope.

The Checksum, Rotation, and NarrowBarWidth properties have no effect on this bar code.

BarCodeWidth Function

This function allows you to retrieve the width of the barcode. Simply fill in the BARCODEPRINTDATA structure with the proper values and call this function. The return value of this function is a integer value that indicates the width of the barcode. If the return value is zero, the function failed. Use the BarCodeGetLastErrorCode or the BarCodeGetLastErrorString functions to retrieve the last error.

This function is declared in the 'BARCODE.BAS' file as:

```
Declare Function BarCodeWidth Lib "BARCODE.VBX" (lpBarCodeData As BARCODEPRINTDATA) As Integer
```

Returns:

Returns the width of the barcode.

See also: PrinterTop, PrinterLeft, and the PrinterHDC properties to print the barcode by setting properties rather than using this function call. See the BarCodePrint function for an example.

BarCodePrint Function

This function allows you to print a bar code on a device context that you have created before calling this function. Simply obtain a device context for the device that you wish to print on and fill in the members of the [BARCODEPRINTDATA](#) structure. The return value of this function is an integer value that indicates the width, in pixels, of the barcode. If the return value is zero, the function failed. Use the [BarCodeGetLastErrorCode](#) or the [BarCodeGetLastErrorString](#) functions to retrieve the last error.

This function is declared in the 'BARCODE.BAS' file as:

```
Declare Function BarCodePrint Lib "BARCODE.VBX" (lpBarCodeData As BARCODEPRINTDATA) As Integer
```

Returns:

Returns the width of the barcode.

The following is an example:

```
Dim mBarCode As BARCODEPRINTDATA
Dim iWidth%

mBarCode.BarCodeType = 0
mBarCode.Checksum = 0
mBarCode.Text = "123456"
mBarCode.Rotation = 0
mBarCode.NarrowBarWidth = 2    ' In Pixels
mBarCode.Ratio = 0
mBarCode.hDC = Printer.hDC
mBarCode.x = 0                 ' In Pixels
mBarCode.y = 0                 ' In Pixels
mBarCode.Height = 50          ' In Pixels

iWidth = BarCodeWidth(mBarCode)    ' returns the width of the barcode, in pixels
BarCodePrint(mBarCode)

Printer.NewPage
Printer.EndDoc
```

See also: [PrinterTop](#), [PrinterLeft](#), and the [PrinterHDC](#) properties to print the barcode by setting properties rather than using this function call.

BARCODEPRINTDATA Structure

In Visual Basic this structure is defined as:

```
Type BARCODEPRINTDATA
  BarCodeType As Integer
  Checksum As Integer
  Text As String * 60
  Rotation As Integer
  NarrowBarWidth As Integer
  Ratio As Integer
  hDC As Integer
  x As Integer
  y As Integer
  Height As Integer
End Type
```

In Visual C++ the structure is as follows:

```
typedef struct tagPrintBarCode {
  int BarCodeType;
  int Checksum;
  Byte Text[60];
  int Rotation;
  int NarrowBarWidth;
  int Ratio;
  int hDCPrinter;
  int x;
  int y;
  int Height;
} *LPBARCODEDATA;
```

BarcodeGetLastErrorCode Function

This function allows you to retrieve the last error code. This function can be used in Visual Basic or Visual C++.

In Visual Basic this function is defined as:

```
Declare Function GetLastErrorCode Lib "BARCODE.VBX" () As Integer
```

```
int BarcodeGetLastErrorCode(void);
```

This function returns the last error code.

In Visual C++ this function can be called by using the 'Load Library', 'GetProcAddress', and 'FreeLibrary' functions. The following is an example:

```
HINSTANCE hVBXInst;  
int TheLastErrorCode;  
int (FAR PASCAL *GetErrorcode) (void) = NULL;  
  
hVBXInst = LoadLibrary("BARCODE.VBX");  
GetErrorcode = (int (__far __pascal *) (void)) GetProcAddress(hVBXInst, "BarcodeGetLastErrorCode");  
if (GetErrorcode != NULL)  
    TheLastErrorCode = (*GetErrorcode)();  
FreeLibrary(hVBXInst);
```

BarcodeGetLastErrorString Function

This function will return a string that describes the last error. This function must not be used in Visual C++ because this function returns a Visual Basic String. See the ['BarcodeGetLastErrorStringC'](#) function.

```
Declare Function GetLastErrorString Lib "BARCODE.VBX" () As String
```

BarcodeGetLastErrorStringC Function

LPSTR BarcodeGetLastErrorStringC(LPSTR Buffer, int BufferLength);

LPSTR *Buffer*; /* pointer to buffer that will hold string */
int *BufferLength*; /* length of buffer */

This function retrieves the last error string into a buffer. In Visual C++ this function can be called by using the 'Load Library', 'GetProcAddress', and 'FreeLibrary' functions. The following is an example:

```
HINSTANCE hVBXInst;  
char Str[81] = "";  
LPSTR (FAR PASCAL *GetErrorStr) (LPSTR, int) = NULL;  
  
GetErrorStr = (char __far *(__far __pascal *) (LPSTR,  
int))GetProcAddress(hVBXInst, "BarcodeGetLastErrorStringC");  
  
if(GetLastErrorString != NULL)  
    (*GetErrorStr)((LPSTR)Str, 81);  
FreeLibrary(hVBXInst);
```

ReadMe

There are two ways to print the barcode using BarCode/VBX. The first way is to print via properties. This involves setting the PrinterTop, PrinterLeft and PrinterHDC properties. The PrinterTop and PrinterLeft properties set the upper left corner of the barcode. When the PrinterHDC property is set with a valid device context handle, the barcode will be printed. The handle to the device context can be the Visual Basic printer, the device context returned by a common dialog box, or one that you have created.

For an example of printing via properties, see the PrinterHDC property.

When using the Action property to copy the barcode to the clipboard, be aware that, if you specify a bitmap, only the portion of the barcode that is visible will be copied to the clipboard.

There are two True-Type fonts included with BarCode/VBX. They are OCR-A and OCR-B. They were installed in the directory specified when the installation was performed. These fonts may be installed by selecting 'Fonts' from the Control Panel.

