Contents

The dBarcode OCX Control

dBarcode OCX is a OLE Control which allows barcode images, in the form of metafile Pictures, to be created within the user's own software. A Picture may be displayed on screen, printed on a printer or passed to the Windows clipboard for incorporation into another Windows application.

Although primarily designed for Microsoft's Visual Basic 4, OCX controls may be employed with other OCX-enabled applications, such as the Visual Basic for Application provided in Microsoft Office 95 and later. Examples provided in this manual are based on Visual Basic 4.

More:

Registering the control

Adding the control to a Visual Basic form

Setting and retrieving property values

Registering the control

Before an OCX control may be used within a particular Windows installation it must be registered with the Windows System Registry. Because this control is provided as a developer's tool, automatic one-time registration is NOT performed by the installation program. However, the installation program does provide programs (and icons) for registering and un-registering dBarcode OCX. If it becomes necessary to register the OCX manually the following commands may be executed from within program manager or as command line commands:

To register the OCX

C:\path\regsvr.exe C:\path\DBCOCX16.OCX for the 16 bit version, or C:\path\regsvr32.exe C:\path\DBCOCX32.OCX for the 32 bit version.

Similarly the OCX may be un-registered using:

C:\path\regsvr.exe C:\path\DBCOCX16.OCX -u for the 16 bit version, or C:\path\regsvr32.exe C:\path\DBCOCX32.OCX -u for the 32 bit version.

If an OCX is moved or upgraded ir must be re-registered.

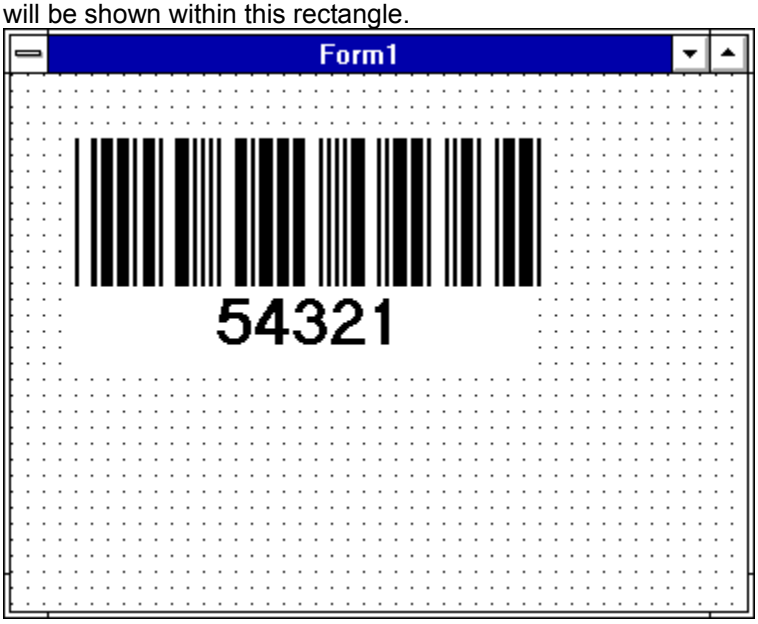
Once registered an OCX may be used within Visual Basic 4 (and other environments). The dBarcode OCX may be included on Visual Basic's Toolbar by checking the

Dbcocx OLE Custom Control

entry in the Custom Controls dialog accessible on the Tools menu. The OCX then appears as an icon on the toolbar.

Adding the control to a Visual Basic form

Clicking on the dBarcode OCX icon causes the cursor to change to a cross. Positioning the cross on a Visual Basic form, holding down the left mouse button and dragging the cursor down and to the left, then releasing the mouse button, causes a rectangle to be drawn on the form. An image of a (default) barcode will be shown within this rectangle.



A single Visual Basic Form may contain any number of dBarcode controls. The first to be added will be called Dbcocx1, the second Dbcocx2, and so on; the names may be changed by the user by modifying the Name property within the Visual Basic Properties box.

The dBarcode OCX Control may be operated entirely by setting or retrieving Property values using the Basic language.

Setting and retrieving property values

Clicking on the dBarcode control on the form when the Properties box is displayed will show the current settings for dBarcode's available properties. Most of these may be edited using the Properties box, or may have their values set from within the user's program by statements of the kind

```
Dbcocx1.Caption=12345
```

dBarcode OCX Control properties which are set AFTER a Picture is created may be retrieved within user's programs by statements of the kind:

```
x=Dbcocx1.ErrorCode
```

Distributing Programs using dBarcode Controls or Libraries

Any program using dBarcode OCX Controls must have access to the dBarcode OCX in file DBCOCX16.OCX (16 bit) and DBCOCX32.OCX (32 bit) .

Any program using dBarcode VBX Controls must have access to the dBarcode Library in file BRMK.VBX.

In addition the dBarcode Controls requires access to the DLL which contains the barcode algorithms, DLSBARn.DLL, where n is the version number, and the corresponding licence file DLSBARn.LIC.

The appropriate VBX or OCX file and the corresponding DLL would have been installed in the dBarcode directory during installation of dBarcode and must be included on any distribution disks containing programs which use the control.

If you distribute programs with an installation system you should install the appropriate VBX or OCX, DLL and LIC files in either your programs .EXE file directory or in the Windows System directory, and ensure that the control is registered in the system registry.

If you intend to redistribute programs containing dBarcode controls and/or libraries for commercial purposes, then you must ensure that you obtain the appropriate licence from DL Technology.

A redistributable version - which does not require the licence file - is available from DL Technology

DBARCODE CONTROLS WILL NOT WORK WITHOUT ACCESS TO THE CORRESPONDING LIBRARY FILE(S).

Creating a barcode Picture within a program

A dBarcode OCX produces its barcode image as a Windows metafile. This metafile is displayed directly in the area you create for the OCX - unless you set the Dbcocx.Visible property to FALSE.

To create a printable Picture object containing the barcode within a program the following steps are required:

1. Set those properties which dBarcode requires to generate the barcode, the minimum of which are

Dbcocx1.Caption - specifies the characters which make up the code
Dbcocx1.CodeType - specifies the barcode type required

2. Paint the control's Picture on the form.

```
If Dbcocx1.ErrorCode = 0 Then
    Form1.PaintPicture Dbcocx1.Picture, 20, 20, i, j
```

3. Printing the Barcode on the Printer

To print the barcode image on the printer use the PaintPicture method of the Printer object, eg.

```
Printer.ScaleMode = 6   ' sets printer scale to mm
Printer.PaintPicture Dbcocx1.Picture, 20, 20, Dbcocx1.ImageWidth,    Dbcocx1.ImageHeight
Printer.NewPage
Printer.EndDoc
```

These steps will now be described in greater detail below.

More:

Step 1. Set the dBarcode properties required.

Step 2 Paint the Picture on the form

Step 1. Set the dBarcode properties required.

The properties which need to be set to generate an image

Dbcocx1.AutoParity	specifies whether a checksum is calculated automatically
Dbcocx1.Caption	specifies the characters which make up the code
Dbcocx1.CodeType	specifies the barcode type required
Dbcocx1.Enabled	must be TRUE
Dbcocx1.ImageHeight	required target height of barcode image (in mm)
Dbcocx1.ImageWidth	required target width of barcode image (in mm)
Dbcocx1.Name	name of dBarcode OCX control
Dbcocx1.Visible	determines whether the control's image is displayed on the form.

In addition, if the barcode is to include an image of the text version of the code, the following properties are significant:

Dbcocx1.ShowText	specifies whether the text is include or not
Dbcocx1.CodeFont	determines the font characteristics of the text font. This property has a number of additional properties: Bold, Italic, Size and Name which specify details of the font. These are specified as, for example, Dbcocx1.CodeFont.Name="Arial"

Many of these properties have default values (see the reference section), so do not require changing if you can make do with the default values. The properties which must be set for you to obtain a barcode are

Dbcocx1.Caption	specifies the characters which make up the code
Dbcocx1.CodeType	specifies the barcode type required

A typical example of setting these properties is:

Dbcocx1.CodeType=8 rem code type is 8 for Code-39 type barcode

Dbcocx1.Caption="123456" rem code is 123456

Step 2 Paint the Picture on the form

To paint the control's Picture on the form use the PaintPicture method, or draw your own bars using the Printers Line method. Using the PaintPicture method the code will look like this:

```
If Dbcocx1.ErrorCode = 0 Then
    Form1.PaintPicture Dbcocx1.Picture, 20, 20, i, j
```

where the (20,20) specifies the coordinates on the form of the top lefthand corner of the barcode image, and i and j represent the required width and height of the image respectively.

The control's Picture may be passed to the clipboard using

```
Clipboard.SetData=Dbcocx1.Picture
```

The Visual Basic Picture object contains a metafile image of the generated barcode. In the event of an error in barcode image generation the image contains the text "??ERROR??", and the ErrorCode property is set to a non-zero value;

The control's Picture may be printed using the PaintPicture method of the Printer object, eg.

```
Printer.ScaleMode = 6   ' sets printer scale to mm
Printer.PaintPicture Dbcocx1.Picture, 20, 20,  Dbcocx1.ImageWidth,  Dbcocx1.ImageHeight
Printer.NewPage
Printer.EndDoc
```

Where the (20,20) refers to the top, left coordinates of the barcode image on the printer's page.

Note that the PrintForm method is no recommended, because this method uses the screen resolution for printing and this is not usually adequate for barcodes.

Using dBarcode in a Container

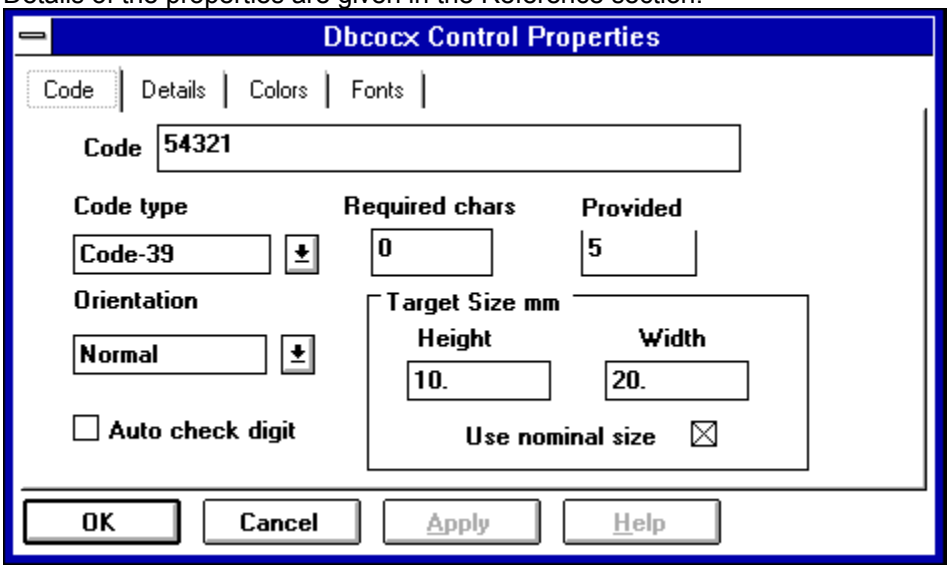
dBarcode may be inserted into Container programs (such as Microsoft Access) generally by selecting Insert OLE Control from the EDIT or INSERT menus and choosing **Dbcocx OLE Control** from the list of options presented.

The properties of the barcode image may be set by selecting the barcode and summoning its Property Pages dialog.

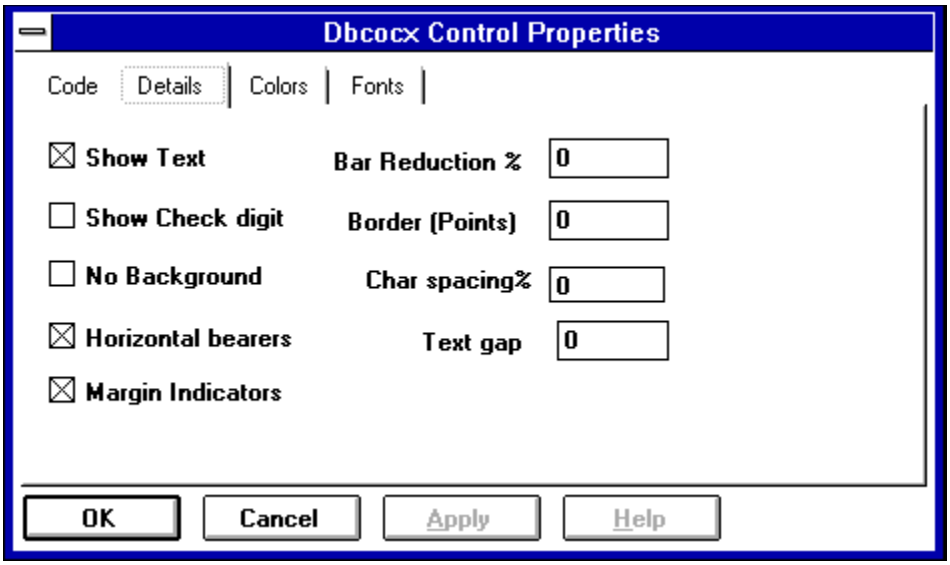
The Property Pages dialog provides a tabbed dialog box containing four pages.

The Code Page provides the user with the opportunity to set the Codetype, the Code value (the Caption), and the target height and width. Checkboxes are provided to allow the Nominal size to be specified (valid only for barcodes which have a nominal size), and for the check digit to be calculated automatically (for those codes which have check digits.

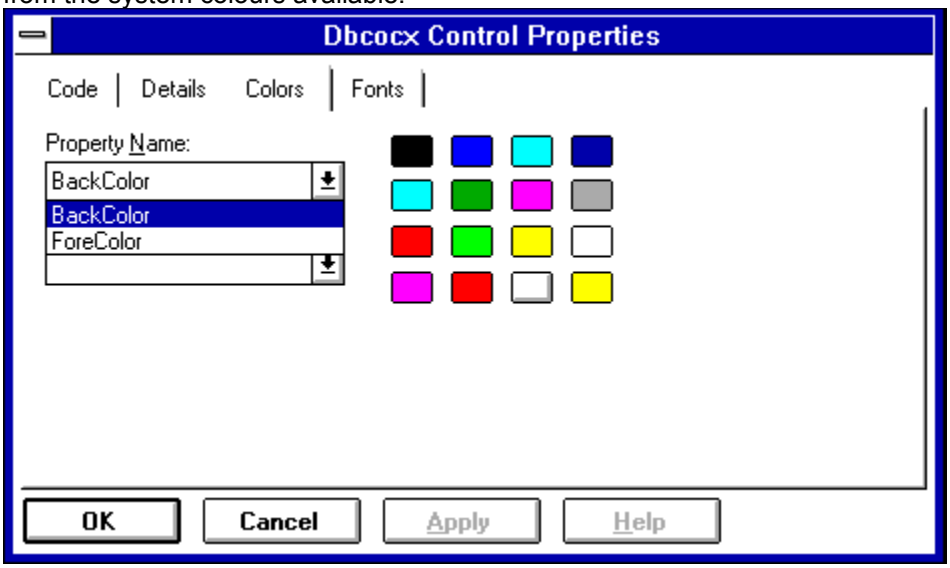
Details of the properties are given in the Reference section.



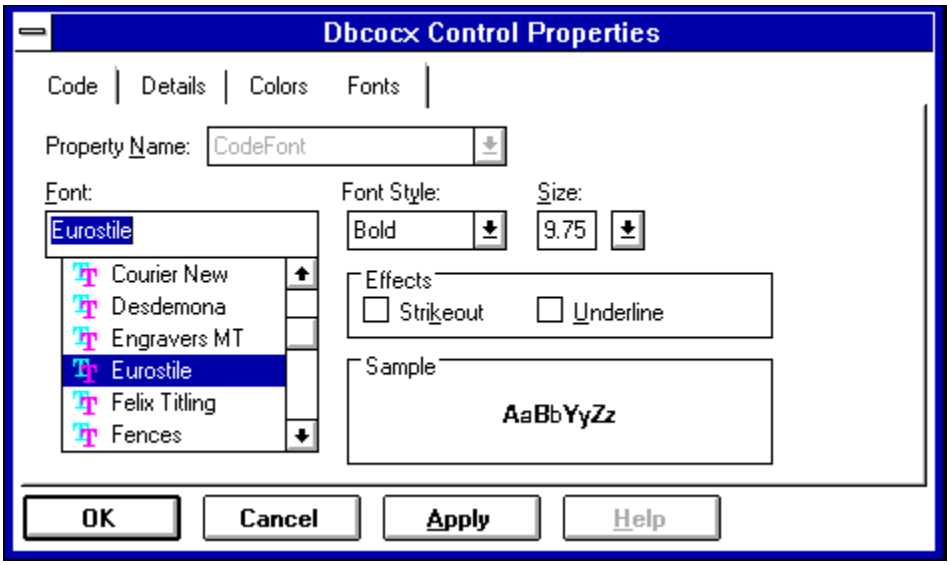
The Details page enables most of the remaining properties to be specified, either through checkboxes or by specifying numberical values in the edit boxes.



The Colours page allows the foreground (bars and text) colour and the background colour to be selected from the system colours available.



The Font page permits the selection of the required text font from a standard Windows font dialog. Note that the size of the font selected will be correctly reproduced only when the metafile is displayed at the size specifed in the target height and width boxes on the Code page. The metafiles image created if fully resizeable, so changing the size of the barcode image will also change the actual size of the font used for the text.



Each page contains an OK button which, if pushed causes the current contents of all the pages to be used in re-creating the barcode image. The pages also contain a Cancel button which may be used to close the Property pages dialog and discard any changes made.

dBarcode OCX Control properties

Dbcocx1.AutoParity	Dbcocx1.BackColor
Dbcocx1.Bearers	Dbcocx1.BorderSize
Dbcocx1.Caption	Dbcocx1.CharSpacing
Dbcocx1.CodeFont	Dbcocx1.CodeType
Dbcocx1.Enabled	Dbcocx1.ErrorCode
Dbcocx1.ForeColor	Dbcocx1.Height
Dbcocx1.ImageHeight	Dbcocx1.ImageWidth
Dbcocx1.Left	Dbcocx1.LightMargins
Dbcocx1.LineReduce	Dbcocx1.Name
Dbcocx1.NominalSize	Dbcocx1.Orientation
Dbcocx1.Parity	Dbcocx1.Pattern
Dbcocx1.Picture	Dbcocx1.Required
Dbcocx1.ShowCheckDigit	Dbcocx1.ShowText
Dbcocx1.TextGap	Dbcocx1.Top
Dbcocx1.Transparent	Dbcocx1.Visible
Dbcocx1.Width	
Dbcocx1.Extra1	Dbcocx1.Extra2

More:

Methods

Property classes

Property details

Methods

Methods

Dbcocx1.CodeInfo(i) Dbcocx1.Refresh

Property classes

The properties which need to be set to generate an image

Dbcocx1.AutoParity	specifies whether a checksum is calculated automatically
Dbcocx1.Caption	specifies the characters which make up the code
Dbcocx1.CodeType	specifies the barcode type required
Dbcocx1.Enabled	must be TRUE
Dbcocx1.ImageHeight	required target height of barcode image (in mm)
Dbcocx1.ImageWidth	required target width of barcode image (in mm)
Dbcocx1.Name	name of the dBarcode OCX control

In addition, if the barcode is to include an image of the text version of the code, the following properties are significant:

Dbcocx1.ShowText	specifies whether the text is include or not
Dbcocx1.CodeFont	determines the font characteristic of the text font Specific characteristics are set as Font properties, eg. Dbcocx1.CodeFont.Name="Arial"

The following properties allow additional control over the image generated:

Dbcocx1.LineReduce	specifies a percentage reduction in the thickness of bars
Dbcocx1.TextGap	specifies percentage of font height by which text should be raised.
Dbcocx1.LightMargins	specifies whether light margin indictaors are to be displayed (EAN/UPC codes).
Dbcocx1.NominalSize	allows the use of the nominal aspect ratio for the specified codetype.
Dbcocx1.Orientation	allows the user to define the orientation of the image produced.
Dbcocx1.ShowCheckDigit	specifies whether any auto checkdigit is to be displayed in text form or not.
Dbcocx1.Transparent	specifies whether the barcode image is created on a transparent background.

The dBarcode VBX Control sets the following properties:

Dbcocx1.Characters:	the number of characters required for barcodes which have a required number, or 0 for barcodes which may be of any length.
Dbcocx1.Parity:	is set to 1 for code types which use a checksum digit, or set to 0 for other codes.
Dbcocx1.Pattern	is set to a Visual Basic string of 0s and 1s, where the 0s represent spaces and the 1s represent bars, each of the minimum width. This property is provided to allow users to draw their own bars.
Dbcocx1.Picture:	the Visual Basic Picture object containing a metafile image of the generated barcode. In the event of an error in barcode image generation the image contains the text "??ERROR??".
Dbcocx1.ErrorCode:	An error code, or 0 if no error occurred.

Property details

BarCoden.AutoParity

Dbcocx1.BackColor

Dbcocx1.Bearers

Dbcocx1.BorderSize

Dbcocx1.Caption

Dbcocx1.CharSpacing

Dbcocx1.CodeFont

Dbcocx1.CodeType

Dbcocx1.Enabled

Dbcocx1.ErrorCode

Dbcocx1.ForeColor

Dbcocx1.Height

Dbcocx1.ImageHeight

Dbcocx1.ImageWidth

Dbcocx1.LightMargins

Dbcocx1.Left

Dbcocx1.LineReduce

Dbcocx1.Name

Dbcocx1.NominalSize

Dbcocx1.Orientation

Dbcocx1.Parity

Dbcocx1.Pattern

Dbcocx1.Picture

Dbcocx1.Required

Dbcocx1.ShowText

Dbcocx1.ShowCheckDigit

Dbcocx1.TextGap

Dbcocx1.Top

Dbcocx1.Transparent

Dbcocx1.Visible

Dbcocx1.Width

Dbcocx1.Extra1 and Dbcocx1.Extra2

BarCoden.AutoParity

Default:	0
Allowed values:	0 (checksum characters not calculated) 1 (checksum characters calculated and appended to code for appropriate code types)

Dbcocx1.BackColor

Default:	&H00FFFFFF&
Allowed values:	0 (black)
to	&H00FFFFFF& (white)

Sets the colour of the image background. This value may be over-ridden by the Transparent property.

Dbcocx1.Bearers

Default: FALSE

Allowed values: TRUE or FALSE
When TRUE causes bearer bars to be draw above and below the barcode.
When FALSE no bearer bars are drawn. Valid only for ITF and ITF-6 code types.

Dbcocx1.BorderSize

Default:	0
Allowed values:	0 (no border)
to	1/3 * ImageHeight

Dbcocx1.Caption

Default: "12345"
Allowed values: Any text string.

Note: only text strings recognised as valid barcodes will result in a barcode picture. Illegal text string will cause an ErrorCode value to be set.

The Caption should be the last property set - as setting this property causes the barcode image to be recreated.

Dbcocx1.CharSpacing

Default:	0
Allowed values	0 and 10 - 100 The percentage of the barcode width which the text under the barcode will take up. A value of 0 causes the characters to be displayed with their "normal" spacing. It is the users responsiblility to ensure that the value provided is sensible in comarison with the fontsize.

Dbcocx1.CodeFont

Default: MS SansSerif
Allowed values: Any accessible TrueType font.

Font details may be specified using the font parameters, eg.
Dbcocx1.CodeFont.Name="Arial"
Dbcocx1.CodeFont.Size=10

Dbcocx1.CodeType

Default: 8
Allowed values: The range of values defined in the barcode types table.

The barcode types table has been moved to the barcodes help file/appendix supplied with the DLSBARn.DLL library which accompanies dBarcode. The barcodes help file will be updated along with the DLL as new barcode types are added.

Dbcocx1.Enabled

Default:	TRUE
Allowed values	TRUE (control enabled) FALSE (control disabled)

Dbcocx1.ErrorCode

Default: Returns a value representing the error code if a valid barcode image cannot be created. Otherwise returns 0.

Read only. Do not set this property.

Dbcocx1.ForeColor

Default:	0
Allowed values:	0 (black)
to	&H00FFFFFF& (white)

Set the colour of the image foreground, i.e. the bars and text colour.

Dbcocx1.Height

Default: value determined when user adds control to a form.

Note this is the Height available for the Control's rectangle. It is not related to the barcode picture. In many cases the Control will be set to invisible, so this parameter may be ignored.

Dbcocx1.ImageHeight

Default:	20 mm
Allowed values:	5 (minimum)
to	form height in mm

The target height of the barcode image. Note that this is the target size of the resizeable metafile. The font height will be as requested if the metafile is reproduced at this size. If the metafile is resized to say double this value then the font aslo be increase in absolute size. Also this value will be placed in the metafile header for those applications which can reproduce metafiles at their specified size.

Dbcocx1.ImageWidth

Default:	25 mm
Allowed values:	5 (minimum)
to	form width in mm

The target with of the barcode image. Note that this is the target size of the resizeable metafile. The font height will be as requested if the metafile is reporduced at this size. If the metafile is resized to say double this value then the font aslo be increase in absolute size. Also this value will be placed in the metafile header for those applications which can reproduce metafiles at their specified size.

Dbcocx1.LightMargins

Default: 1	A value of 1 causes the light margin indicators recommended for some EAN barcodes are to be shown on the image, and causes the Prefix code and checkdigit for UPC-A and UPC-E codetype to be displayed in the light margins. A value of 0 prevents the display of the light margin indicators.
Allowed values	0 and 1

Note that for ITF files specifying LightMargins=1 causes the border bars between the extremes of the bearer bars to be drawn. The required light margin distances are always included in ITF codes.

Dbcocx1.Left

Default: value determined when user add control to a form.
Note this is the Left position available for the Control's rectangle. picture.

Dbcocx1.LineReduce

Default: 0 — the thickness of each line drawn on the barcode image is reduced by this percentage amount. This property may be used to compensate for ink spreading during wet-ink printing.

Allowed values: 0 - 50 (%)

Dbcocx1.Name

Default: Dbcocx*n*
Allowed values: Any name valid for a Visual Basic object.
Design time only.

Dbcocx1.NominalSize

Default: 0	When this property is set to 1 the width of the image created will be scaled to generate the aspect ratio specified for the nominal size image of the current barcode type. When this property is 0 no scaling is applied.
Allowed values:	0 and 1

Note that this feature is relevant only for some EAN, UPC and ITF code types.

Dbcocx1.Orientation

Default: 0	The value of this parameter determines the orientation of the barcode image created.
Allowed values:	0 = normal orientation 1 = image rotated left by 90 degrees (clockwise) 2 = image rotated right by 90 degrees (anti-clockwise) 3 = image inverted.

Note that the rotation of text is only supported by TrueType and other rotatble fonts. Note also that some applications do not correctly handle metafiles which contain rotated text.

Dbcocx1.Parity

Default:: return 1 for code types which allow the inclusion of a checksum, otherwise returns 0.

Note that this property returns 0 for EAN 128 and Code 128 in spite of the fact that these codes define checksum digits. This is because these codes cannot be used without the check digits and these digits are **always** calculated by the control and are not under user control. Check digits for these codes are **not** displayed.

Read only. Do not set this property. The value of this parameter can be determined following the setting of CodeInfo to the code type, eg.

```
if Dbcocx1.CodeInfo(1)>=0 then
x=Dbcocx1.Parity
```

Dbcocx1.Pattern

Default: A Visual Basic string containing a pattern of 0s and 1s, where 0 represents a space of minimum width and a 1 represents a bar of minimum width. Within Visual Basic the 1s may be used to indicate that a bar (filled rectangle) must be drawn.

Read only. Do not set this property.

Dbcocx1.Picture

Default: The Picture of metafile type returned by the dBarcode OCX control. This contains the image of the barcode and (if enabled) its text.

Read only. Do not set this property.

Dbcocx1.Required

Returns: 0 or the number of characters required for a valid barcode of the current type. 0 indicates any number of characters permitted.

Read only. Do not set this property. The value of this parameter can be determined following the setting of CodeInfo to the code type, eg.
if Dbcocx1.CodeInfo(1)>=0 then
x=Dbcocx1.Required
The value returned depends on the current setting for AutoParity - ie. the value specifies the number of characters that need to be provided by the user.

Dbcocx1.ShowText

Default:	0
Allowed values	0 (text version of the code is NOT included in the barcode image) 1 (text version of the code IS included in the barcode image)

Dbcocx1.ShowCheckDigit

Default: 0 — When set to 1 this property causes the any automatically calculated check digit to be included in the any text displayed along with the barcode. When set to 0 the check digit is not displayed.

Note that this property has no effect on those codetypes for which the checkdigit display is mandatory (including EAN and UPC codes).

Dbcocx1.TextGap

Default: 0	A value given as a percentage of the text font point size by which text is raised within its bounding rectangle.
Allowed values:	0-100 (%)

Note: For most applications this value should be set to 0.

Dbcocx1.Top

Default: value determined when user adds control to a form.

Note this is the Top position available for the Control's icon. It is not related to the barcode picture. In most case the Control will be set to invisible, so this parameter may be ignored.

Dbcocx1.Transparent

Default:	FALSE
Allowed values	TRUE or FALSE. When set to TRUE this property causes the barcode image to be created on a transparent background (ie. and background colour is ignored). The property should be set to 1 only by users who are confident that the final barcode image will be produced on a background which does not interfere with barcode scanning.

Dbcocx1.Visible

Default:	TRUE
Allowed values	TRUE (the dBarcode VBX icon is displayed on the form at run time) FALSE (the dBarcode VBX icon is NOT displayed on the form at run time)

Dbcocx1.Width

Default: value determined when user adds control to a form.

Note this is the Width available for the Control's icon. It is not related to the barcode picture. In most case the Control will be set to invisible, so this parameter may be ignored.

Dbcocx1.Extra1 and Dbcocx1.Extra2

Default 0

These additional properties are not normally used. However, they do provide additional functions for a limited number of specific barcode types. See Barcodes HELP for details.

Methods

Dbcocx1.CodeInfo(i)

Dbcocx1.Refresh

Dbcocx1.CodeInfo(i)

Parameter: i 0 - maximum codetype value (currently 37)
Returns i if call is successful, or -1 otherwise.
Calling CodeInfo with the parameter i returns infomation about the code type of that value, including the Name, ImageHeight, ImageWidth, Required number of Characters, and Parity (whether Check digits are supported).
Run time only. This parameter is useful for determining the range of barcode types supported by the library. Unsupported types return -1.

Dbcocx1.Refresh

Causes the barcode image to be recreated. If the Control is visible it will be redrawn. Note that a Picture derived from the control's Picture property is not redrawn by the Refresh method. If you need to recreate the picture then do something with the Dbcocx1.Picture property - such as use PaintForm, or copy it to the clipboard.

If you are using the control to create a metafile without the control's image being visible on a form, then you also need to access the pricture property using a instruction such as

```
x = Dbcocx1.Picture
```

Error codes

The Property Dbcocx1.ErrorCode AFTER Dbcocx1.Calc has been set to 1 may contain an error code if the dBarcode VBX Control was unable to generate a barcode Picture. The property value will be one of the following:

0 No error
1 Wrong code length
2 Unrecognised code type
3 Wrong add-on code length
4 Illegal character in code
5 Error in embedded code
6 Generated line width less than 1 unit

SISAC codes

[Note: Requires dBarcode-SISAC library]

Unlike most other barcodes the SISAC barcode symbol does not have a one-to-one correspondence with the SICI code printed underneath it. dBarcode SC generates the SISAC barcode from the SICI code, and it can only do this if the SICI code itself is correct. If the SICI code is not correct then the text version of the SICI code will appear in the dBarcode window, but the barcode symbol will not.

The SICI code must be entered into the Code edit box, and it **must contain at least the following items:**

The ISSN number complete with a hyphen between digits 4 and 5, e.g. 1234-5678

A date item enclosed in brackets. If no date item is required the () symbols MUST STILL BE PRESENT.

A number item is optional. e.g. 14:1

Index or supplement numbers are optional, e.g. *1

The standard version number which is currently ;1- and all three characters MUST BE PRESENT

A SICI check digit may immediately follow the - of the version number. The check digit may be entered manually or may be calculated by dBarcode by enabling Auto Checksum on the Options menu.

Note: When copying SICI codes from publications it is not always easy to distinguish the : and ; characters. SICI codes ALWAYS end with (semicolon) ;1-n where n is a check digit.

dBarcode does not currently support SICI location codes.

Notes on Metafiles

The picture images placed on the clipboard by dBarcode are ANISOTROPIC metafiles. This means that they can be resized within applications (usually by dragging a corner).

While the barcode bars can be resized over very wide ranges, any text included within the image may not resize as expected. In general changing the height of the image by resizing within another application will change the fontsize used to render the text. Changing the width of the image within another application may cause the position of any text under the barcode to change.

To overcome text size problems caused by resizing metafile images choose an alternative fontsize within dBarcode. The use of TrueType fonts is recommended to prevent unusual effects caused by resizing of text.