

# Contents

## Introduction

[dBarcode - Barcodes for Windows](#)

## Using dBarcode

[Creating a barcode image](#)

[Copying a barcode image to the Windows Clipboard.](#)

[Saving a barcode image as a WMF file](#)

[Notes on Metafiles](#)

## Reference

[The dBarcode Window](#)

[Menu Reference](#)

[Command Reference](#)

[Dialogs](#)

## OLE

[Embedded Objects](#)

## dBarcode - Barcodes for Windows

dBarcode is a Windows utility which enables barcode images to be created and inserted into other Windows applications.

dBarcode images are normally created as Windows Metafiles. These are continuously scalable and maintain a high quality at any size. However, dBarcode can also create bitmaps and save them as BMP, PCX or Encapsulated PostScript files. The 32 bit version of dBarcode will also create Enhanced Metafiles, which behave differently to standard metafiles on resizing. (See the Notes on Metafiles section.)

While most barcodes are used as black bars on a white background, dBarcode does provide the option of selecting both foreground (bar) and background colours, and dBarcode can display the code text below the barcode using a font of your choice.


dBarcode may be used as a standalone program for creating barcode images (which can then be copied to the clipboard for pasting into other applications, or saved into files); the Professional version may also be used as an OLE server for OLE enable applications.

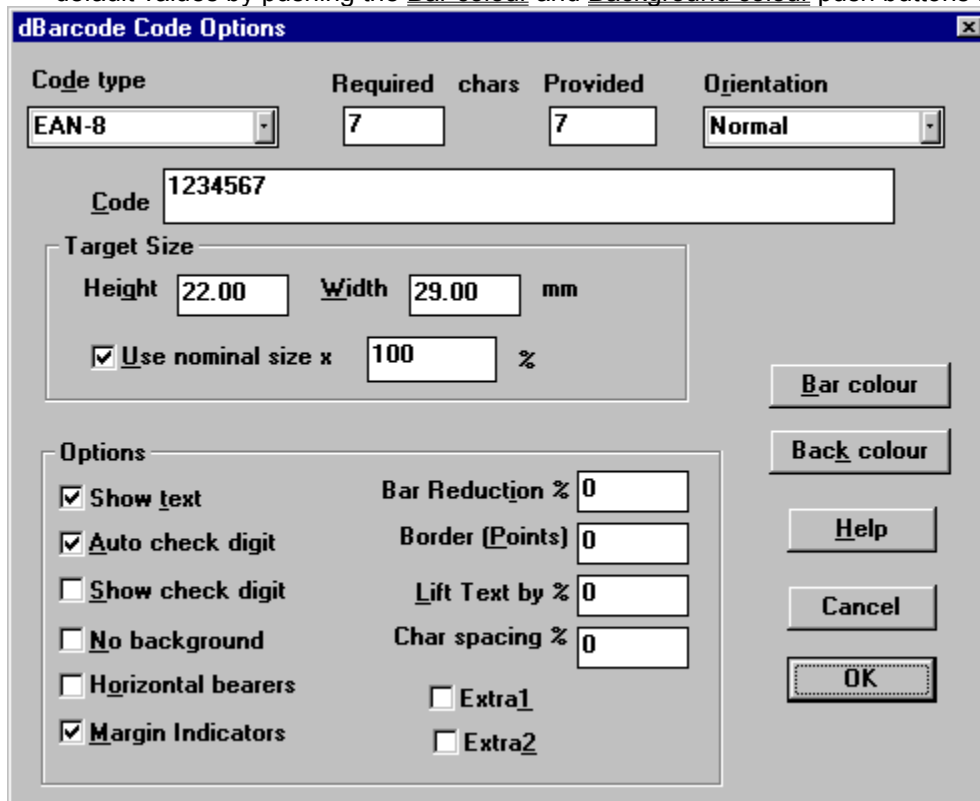
Please note that before dBarcode Pro may be used as an OLE server it must be run at least once as a standalone program so that it can register itself with the Windows Registry.



## Creating a barcode image

The procedure for creating a barcode image is simple:

1. Select the Code Style item on the Options menu or on the control bar to display the Code style dialog.  

2. Choose the type of barcode from the Code Type combo box.
2. Check that the Auto Checkdigit checkbox is set as required.  
The checkbox is checked if dBarcode's automatic checksum calculation facility has been enabled.  
The number of characters required in a barcode may depend on whether an automatic checkdigit calculation is to be performed.
3. Enter the required code into the code box.  
The Characters required box specifies how many characters are required for the chosen barcode type. If 0 is displayed then any number of characters may be entered. The Characters provided box display the number of characters currently entered into the code box.
4. If you wish the code to be displayed as text under the barcode then the Show Text checkbox must be checked.
5. The Foreground and Background colours used for the barcode image may be changed from their default values by pushing the Bar colour and Background colour push buttons respectively.



**dBarcode Code Options**

Code type	Required chars	Provided	Orientation
EAN-8	7	7	Normal

Code: 1234567

Target Size  
Height: 22.00 mm, Width: 29.00 mm  
 Use nominal size x 100 %

Options

<input checked="" type="checkbox"/> Show text	Bar Reduction %: 0
<input checked="" type="checkbox"/> Auto check digit	Border (Points): 0
<input type="checkbox"/> Show check digit	Lift Text by %: 0
<input type="checkbox"/> No background	Char spacing %: 0
<input type="checkbox"/> Horizontal bearers	<input type="checkbox"/> Extra1
<input checked="" type="checkbox"/> Margin Indicators	<input type="checkbox"/> Extra2

Buttons: Bar colour, Back colour, Help, Cancel, OK

6. The thickness of any border around the barcode, may be set (in points) in the Border box as required. Any border is shown using the background colour.
7. Push the OK button. The barcode image will be displayed if the code is valid; otherwise an error.

message is displayed in the status bar.

**Notes:**

If the Status bar display shows an error this is because the code you have entered may not be legal for the codetype you have selected. This may be because the number of characters is not correct, or because you have included spaces or illegal characters (such as alphabetic characters in a numeric coding scheme, or characters in the wrong case). The Status bar will show the reason why the image has not been created. Check the code in the code box and try again.

If no image is visible it may be because the foreground and background colours have been set to the same colour.

If you wish to create a barcode image which will occupy a particular size, then adjust the Target Size values in the Code Style dialog and enter the required height and width in millimetres.

If you wish to copy the barcode image to the clipboard choose Copy from the Edit menu or push the copy button on the control bar.

**More:**

Barcode types

## **Barcode types**

Details of the types of barcodes supported are contained in the Barcodes HELP system which accompanies dBarcode. This HELP system is updated at the same time as the barcode library.

## Copying a barcode image to the Windows Clipboard.

The barcode image displayed within the dBarcode window may be copied to the clipboard by selecting the Copy item of the EDIT menu, or by pressing <Ctrl><C>, or by pushing the Copy button on the control bar.



The image placed on the clipboard is a Picture (metafile). This may be pasted into most Windows applications which accept metafile Pictures.

Copying an image to the clipboard replaces any previous image stored on the clipboard.

## **Saving a barcode image as a WMF file**

The barcode image displayed within the dBarcode window may be saved as a Windows Placeable Metafile (.WMF) by selecting Save Image from the FILE menu.

The Windows Save file dialog appears, prompting the user to enter a file name, and, if necessary, to select an appropriate drive and directory in which to save the file.

The WMF files may be loaded directly into applications such as Word for Windows.

### **More:**

[Saving a barcode images as bitmap, PCX or Encapsulated PostScript files.](#)

## **Saving a barcode images as bitmap, PCX or Encapsulated PostScript files.**

Barcode images may be saved as bitmap images in either Windows bitmap files (with the extension BMP), standard PCX files (extension PCX) or in Encapsulated PostScript files (extension EPS), by selecting the required file type from the File Save dialog.



## 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 textsize 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.

Using the Target Size dialog and setting height and width values for the final image is useful in ensuring that an appropriate fontsize is used for the image. The dBarcode window will display the image as it would appear at the target size - although the actual size of the image in the dBarcode window will always be as large as possible.

Note that those barcodes which have a nominal size (eg the EAN-13 and EAN-8 codes) may be placed on the clipboard as metafiles with the correct height to width ratio by checking the Nominal Size checkbox in the Code Style dialog. When this checkbox is checked dBarcode will set the Target Height and Width values to those used for the 100% size of the barcode. Note that most barcodes which have nominal sizes allow the size to be varied in the range 80 - 120% of the nominal values.

Some older Windows applications do not correctly interpret Windows metafiles, and may align text incorrectly - generally placing the text too low on the image. To allow dBarcode's metafiles to be used with such applications, the vertical position of text may be adjusted vertically by a percentage of the font height - see the Code Style dialog for details.

### **More:**

[Enhanced metafiles](#)

[Bitmap images](#)

## Enhanced metafiles

Most modern 32 bit applications support Enhanced metafile pictures, which generally are more easily handled than standard metafiles. The 16 bit version of dBarcode creates standard metafiles, so if you are pasting a clipboard image into a 32 bit application you should ensure that it treats the image as a standard metafile picture. If in doubt you can usually select a menu item such as Paste Special which will offer a choice Picture type. A standard metafile image may appear distorted when treated by an application as an enhanced metafile. This is not a fault with dBarcode - its caused by the other application failing to correctly identify the metafile type. dBarcode 16 does not produce enhanced metafiles because many older applications cannot interpret them correctly.

The 32 bit version of dBarcode supports Enhanced Metafiles in addition to standard metafile pictures. Using the Copy Options of the Edit menu you can select which formats are used when a barcode image is copied to the clipboard. By default, dBarcode 32 will copy both standard and Enhanced metafiles (and bitmaps) to the clipboard, allowing the target application to select the optimum format during a paste operation.

TrueType fonts scale more smoothly in Enhanced metafiles than in standard metafiles - retaining the shape of the original font outline. dBarcode 32 displays the enhanced metafile image on screen by default. If you wish to view the standard metafile image uncheck the Enhanced metafile option of the View menu.

## Bitmap images

Unlike metafiles, bitmaps maintain the resolution with which they were created, so that although a bitmap image may be resized at will, the individual dots of the bitmap also get resized. When dBarcode saves a bitmap image it does so with the resolution of the screen image being displayed at the time. Consequently to obtain the highest possible bitmap resolution the dBarcode Window should be maximized before the bitmap is saved. File space may be reduced by working with a lower window size, but the resolution of the final image will also be reduced.

The user will need to experiment with the size of dBarcode Window to use in order to obtain the optimum resolution required for the kind of barcode image being created, the resolution of their Windows screen (eg. 640\*480, 800\*600, etc.) and the type of printer on which the final image will be printed (eg. 8-pin matrix, 300 dpi, 1200 dpi, etc.). ALL barcode images should be tested in their final form before extensive print runs are undertaken.

Encapsulated PostScript files will contain the target size information just as the metafiles do, but BMP and PCX files will contain the size information of the image visible on screen when the files were saved. (Of course BMP and PCX images may be reduced in size when imported into other other applications.)

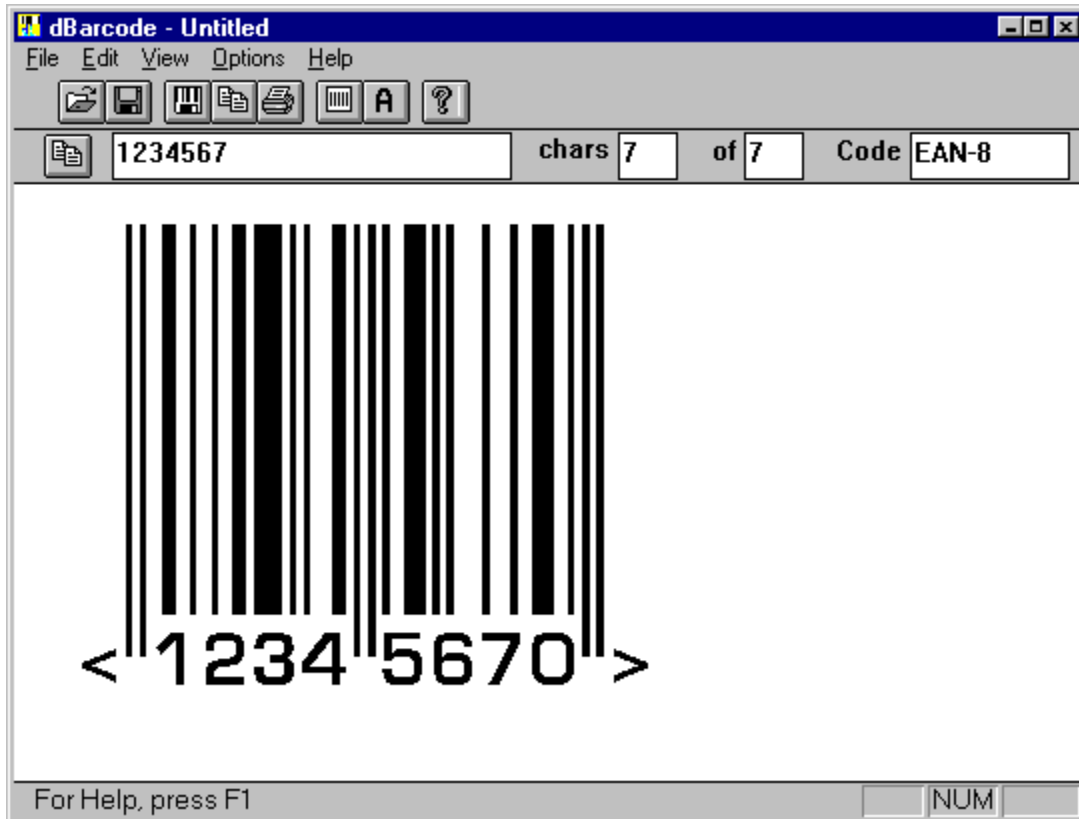
## The dBarcode Window

The dBarcode window contains five areas of importance:

The Menu bar The Tool bar

The Control bar The Image area

The Status bar



### More:

[Menu bar](#)

[The Tool bar](#)

[Control bar](#)

[Image area](#)

[Status bar](#)

## Menu bar

The Menu bar contains the following items:

- File Provides file opening, saving and printing facilities
- Edit Permits the barcode image to be copied to the clipboard
- View Display or Hide the Control, Tool and Status bars
- Options Allows the image properties to be modified
- Help Provides access to the HELP system

Clicking on any item will cause the display of a pull-down menu which offers a list of related commands which may be given to dBarcode.

## The Tool bar

The Tool bar contains icons which mimic the effect of selecting commonly used menuitems.



From left to right the icons are equivalent to:

Open file

Save file

Save Image

Copy

Print

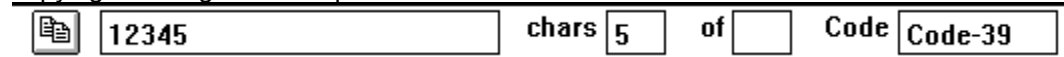
Code Style

Font

Help

## Control bar

The Control bar shows information related to the image displayed in the Image area, and provides the user with a rapid means of modifying the most frequently changed item of data (the barcode code) and copying the image to the clipboard.



The Control bar contains the following items:

The Copy button

A Code box

Provided characters box

Required characters box

Code type box

### **More:**

[Copy button](#)

[Code box](#)

[Provided characters box](#)

[Required characters box](#)

[Code type box](#)

## **Copy button**

Pushing this button copies the current barcode image to the clipboard. The effect is identical to choosing Copy from the Edit menu.



## **Code box**

This is a standard edit box into which the characters which will be used to create the barcode must be entered by the user. The usual Windows editing actions may be undertaken in this box (ie. <Ctrl><Ins> copies any selected text to the clipboard; <Shift><Ins> pastes any text on the clipboard into the box at the current cursor position).

## **Provided characters box**

This shows the number of characters currently displayed in the Code box.

## **Required characters box**

This shows the number of characters required for the currently selected code type. Codes which allow any number of characters display 0 in this box.

## Code type box

This shows the current barcode type. The barcode type can be changed only within the Code Style dialog.

## Image area

The Image area is that portion of the dBarcode window in which the barcode image is displayed. The Image area is resized automatically as the dBarcode window is resized, and cannot otherwise be altered. The image area will be displayed using a size which is the best fit within the dBarcode window, while maintaining the height to width ratio specified in the Target Size settings.

When a background colour is chosen, then the image area is painted with the background colour before the barcode image is drawn using the foreground colour. If a border size is provided, then that size represents the distance (in printers' Points) which will be left between the edge of the barcode picture and the edge of the image area.

When a Copy operation is performed the whole image area is copied to the clipboard.



## Status bar

This bar displays a status message which is generated when dBarcode attempts to create a barcode image. For a correctly formed barcode the status bar should display the message "OK". When a barcode cannot be created the status bar will contain an error message. The possible status messages are:

**More:**

[Error messages](#)

## **Error messages**

*code*    *message*

- 0        OK (ie 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
- 7        font error
- 8        unable to create metafile
- 9        no code provided

Note that the status bar also displays context-sensitive information as the mouse is moved over areas of dBarcode's window.

# Menu Reference

[File Menu](#)

[Edit Menu](#)

[View Menu](#)

[Options Menu](#)



## File Menu

The File menu includes commands that enable you to open and save a dBarcode configuration, print the image or exit.

<u>Open</u>	Open a named configuration file.
<u>Save</u>	Save the current configuration.
<u>Save As</u>	Save the current configuration as a named file.
<u>Save Image</u>	Save the current image as a WMF file.
<u>Print</u>	Print the current image.
<u>Print Preview</u>	Display a WYSIWYG view of what would be printed by a Print command.
<u>Print Setup</u>	Summons the Windows printer setup dialog.
<u>Exit</u>	Close dBarcode.

## Edit Menu

The Edit menu contains commands that enable you to copy the current Barcode image to the Windows Clipboard in a variety of formats.

Copy Copy Image to clipboard

Copy Options (32 bit version only) Displays the Copy Options dialog.

## View Menu

The View menu includes commands that enable you to display or hide the Tool bar, the Control bar and the Status bar. The display of each item may be toggled on or off by clicking on the required menuitem.

The menu also contains a Redraw command, which may be used for refreshing the window display if this becomes corrupted.

Tool bar checked when enabled

Control bar checked when enabled

Status bar checked when enabled

Redraw redraw the image

Enhanced metafiles (32 bit version only) Checked when an Enhanced metafile is being viewed.

Unchecked when a standard metafile is being viewed.

## Options Menu

The Options menu includes commands that enable you to modify the Barcode image.

- Code Style Summons the Code Style dialog box.
- Font Summons the Font dialog to choose the Font used for displaying text under the barcode.
- ShowText Toggle on/off the display of text under the barcode.
- Save Defaults Save the current configuration as the default configuration.

# Command Reference

[File Open Command](#)

[File Save Command](#)

[File Save As Command](#)

[File Save Image Command](#)

[File Print command](#)

[File Print Preview command](#)

[File Printer Setup command](#)

[File Exit Command](#)

[Edit Copy Command](#)

[Edit Copy Options command](#)

[View Toolbar command](#)

[View Controlbar command](#)

[View Statusbar command](#)

[View Redraw command](#)

[View Enhanced metafiles command](#)

[Options Code Style Command](#)

[Options Font Command](#)

[Options Show Text Command](#)

[Options Save defaults Command](#)

## **File Open Command**

Displays the Open file dialog box, which allows the user to select a configuration file for opening. The dialog box shows the list of configuration file names available in the current directory, using the default extension .BRC.

An alternative drive and/or path may be selected from the drive and directory combo-boxes.

## **File Save Command**

Save the current configuration in the current file.

The name of the current file is displayed in dBarcode's title bar. If there is no current file the title bar shows (untitled), and Save behaves as Save As.

## **File Save As Command**

Displays the Save file dialog box, which allows the user to name a file for saving the current configuration.

An alternative drive and/or path may be selected from the drive and directory combo-boxes.

Configuration files are automatically saved with the extension .BRC unless an alternative extension is explicitly included in the filename.



## **File Save Image Command**

The barcode image displayed within the dBarcode window may be saved as a Windows Placeable Metafile (.WMF) by selecting Save Image from the FILE menu.

The Windows Save file dialog appears, prompting the user to enter a file name, and, if necessary, to select an appropriate drive and directory in which to save the file.

The WMF files may be loaded directly into applications such as Word for Windows.

## **File Print command**

Prints the current image on the currently selected printer at the currently specified target size.

The current printer may be changed in the Printer setup dialog (see below), and the target size may be changed in the Code Style dialog (see below).

Note that printing facilities are provided for proofing purposes only.

## **File Print Preview command**

Display a WYSIWYG view of what would be printed by a Print command.

## **File Printer Setup command**

Summons the Windows [Printer Setup dialog](#) which allows the target printer to be selected, and attributes of the printer (eg. page orientation etc.) to be modified.

## **File Exit Command**

Once selected dBarcode terminates. If any changes have been made to the current image configuration, the user will be prompted to save such changes before dBarcode terminates.

## **Edit Copy Command**

When the Copy menuitem is chosen the current barcode image area displayed in dBarcode is copied to the clipboard using the formats specified in the Copy Options dialog.

The Image on the clipboard may be viewed using a clipboard viewer, such as the CLIPBRD.EXE program supplied with Windows. The image may be pasted into most applications which can accept pictures.

Note that copying an image to the clipboard automatically erases the clipboard of any other information on the clipboard.

## **Edit Copy Options command**

(32 bit versions only)

The Copy Options command causes the Copy Options dialog to be displayed. This allows the formats which will be copied to the clipboard during a Copy command to be selected.

## **View Toolbar command**

Choosing this menuitem toggles the display of the dBarcode toolbar. The Toolbar display is enabled when this menuitem is ticked.



## **View Controlbar command**

Choosing this menuitem toggles the display of the dBarcode control bar. The control bar display is enabled when this menuitem is ticked.

## **View Statusbar command**

Choosing this menuitem toggles the display of the dBarcode status bar. The status bar display is enabled when this menuitem is ticked.

## **View Redraw command**

Choosing this item cause dBarcode's window to be redrawn.

## **View Enhanced metafiles command**

(32 bit versions only)

When this item is checked displays on screen the Enhanced metafile form of the barcode image. When the item is unchecked, the standard metafile form is displayed. Clicking on the menuitem will change the check state.

## **Options Code Style Command**

When selected the dBarcode Code Style dialog box is displayed. This allows the user to specify most of the parameters associated with the barcode image generation. For details see the Code Style dialog.

## **Options Font Command**

When selected, the Font dialog box is displayed and allows the choice of a font which will be used to render the barcode text (if enabled) when the barcode image is redrawn.

The barcode image is automatically redrawn when the dialog is closed.

Note that only TrueType or similar rotatable fonts may be used where text is to be included in a rotated image.

## **Options Show Text Command**

When ticked, dBarcode display a text version of the code below the barcode image. When not ticked, the text version is not displayed.

Changing between ticked and unticked may be accomplished by selecting this menuitem.

## **Options Save defaults Command**

When selected the current configuration is saved within the DBARCODE.INI file located in the Windows directory. This configuration is then the default configuration which is loaded automatically and used each time dBarcode is started.



# Dialogs

[Code Style dialog](#)

[Colour dialog](#)

[Copy Options dialog](#)

[File Open dialog](#)

[Font selection dialog](#)

[Print Setup dialog](#)

[File Save dialog](#)

## Code Style dialog

The Code Style dialog is the primary mechanism for specifying details of the barcode image to be generated.

The screenshot shows the 'dBarcode Code Options' dialog box. It has a title bar with a close button. The main area is divided into several sections:

- Code type:** A dropdown menu showing 'EAN-8'.
- Required chars:** A text box containing '7'.
- Provided:** A text box containing '7'.
- Orientation:** A dropdown menu showing 'Normal'.
- Code:** A text box containing '1234567'.
- Target Size:** A group box containing:
  - Height:** A text box with '22.00'.
  - Width:** A text box with '29.00'.
  - mm:** A label.
  - Use nominal size x**  **%**
- Options:** A group box containing:
  - Show text**
  - Auto check digit**
  - Show check digit**
  - No background**
  - Horizontal bearers**
  - Margin Indicators**
  - Bar Reduction %**
  - Border (Points)**
  - Lift Text by %**
  - Char spacing %**
  - Extra1**
  - Extra2**
- Buttons:** A vertical stack of buttons on the right side: 'Bar colour', 'Back colour', 'Help', 'Cancel', and 'OK' (which is highlighted with a dashed border).

### More:

[Code type combo box](#)

[Code box](#)

[Target Size box](#)

[Orientation combo box](#)

[Show Text checkbox](#)

[Auto Checkdigit check box](#)

[Show Checkdigit check box](#)

[No Background checkbox](#)

[Horizontal bearers checkbox](#)

[Margin indicators checkbox](#)

[Border edit box](#)

[Bar reduction edit box](#)

[Lift Text by edit box](#)

[Character spacing edit box](#)

[Background colour button](#)

Bar colour button

Other options

## **Code type combo box**

This drop-down combo box lists the code types supported by dBarcode. Clicking on the arrow alongside the box causes the list to drop down.

A specific code type may be selected by clicking on the name within the list.

## **Code box**

The code edit box provides the entry point for the user to enter the code which is to be converted into the barcode pattern of bars.

This is a standard Windows edit box, and all the usual editing shortcuts may be used.

## **Target Size box**

The Target Size box contains two edit boxes which allow the user to enter height and width values (in millimetres) for the final image. Note that the metafile image produced by dBarcode may be resized independently of the target size, but that when resized any fontsize will also change.

Note that those barcodes which have a nominal size (eg the EAN-13 and EAN-8 codes) may have the nominal size entered into the height and width edit boxes automatically, by checking the Use Nominal Size checkbox. Barcodes may be scaled to a percentage of the nominal size by entering the required value in the nominal size multiplier edit box. Allowed values are 10 - 200%. Note, however, that the acceptable size range for most barcodes types which have nominal sizes is 80-120%.

## **Orientation combo box**

The orientation combo box contains a drop down list of the allowed orientations which dBarcode supports. Clicking on the arrow alongside the box causes the list to drop down.

A specific orientation may be selected by clicking on the name within the list.

Note that some older application do not correctly handle the positioning of text which has been rotated. Also only TrueType or similar rotatable fonts may be used where text is to be included in a rotated image.

## **Show Text checkbox**

When this option is checked dBarcode will include the text form of the barcode under the barcode image - using the currently selected font properties to render the text.



## **Auto Checkdigit check box**

When this option is checked dBarcode automatically calculates the checksum digits of the code. Consequently checksum digits should not be entered by the user when this item is checked. dBarcode will modify the number which appears as Characters Required when the Auto Checkdigit option is checked.

Note that for Code 128 and EAN-128 the checksum calculation is always automatic and the resulting characters are not displayed in human readable form. These features are specified by the EAN-128 standard.

## **Show Checkdigit check box**

When this option is checked dBarcode will include checkdigits which it calculates automatically with the text version of the code displayed below the bars. When the option is not checked, such check digits will not be displayed.

Note that the display of check digits is NOT optional for EAN and UPC codes.

Note also that some codes (eg. Code 39) may generated alphabetic check digits from numeric codes.

## **No Background checkbox**

When this option is checked the background is set to Transparent and no background (or border) colour is applied to the image.

Note however, that this facility should be used with caution. Images showing through a barcode can disrupt the scanning process.

## **Horizontal bearers checkbox**

When this option is checked horizontal bearer bars are included on ITF barcodes. The ITF specification does indicate that the horizontal bearer bars should be printed wherever possible.

## **Margin indicators checkbox**

When this item is checked margin indicators are included for EAN and ITF barcodes.

For EAN codes the margin indicators are < and > symbols, while for ITF codes the margins are indicated by thick bars connecting the ends of the horizontal bearer bars.

In both cases the specifications recommend the use of margin indicators, but their use is not mandatory.

## **Border edit box**

The Border edit box allows the choice of a thickness (in Points) of a border between the barcode image and the edge of the image area. Any border is displayed using the current background colour.

The permitted values of the border thickness is in the range 0 - 72 Points (0 - 1 inch).

**Bar reduction edit box**

The user may enter into this box a percentage reduction which is used to reduce the thickness of all lines in the barcode. Any value between 0 and 50 may be entered. This facility may be used to compensate for ink spread following wet ink printing.

### **Lift Text by edit box**

The user may enter into this box a percentage amount (of the font height) by which text may be moved up. This provides a correction for some older applications which do not correctly align text in metafiles



## **Character spacing edit box**

The value in the Character spacing edit box determines how characters are spaced out under a barcode. A value of 0% results in the characters being centred under the barcode with their normal default spacing. A value in the range 10 - 100% causes dBarcode to separate the characters from one another so that the length of the text occupies the specified percentage of the barcode length.

Note that some applications do not correctly support alignment of characters within metafiles.

## **Background colour button**

When pushed, the Colours dialog box is displayed and allows the choice of a colour with which the image background will be rendered when the barcode image is redrawn.

## **Bar colour button**

When selected, the Colours dialog box is displayed and allows the choice of a colour with which the image bars and text font will be rendered when the barcode image is redrawn.

## **Other options**

Two additional option checkboxes have been added to the dialog, Extra1 and Extra2. These will normally be left unchecked. They do, however, provide for extra functions which are specific to particular barcode types. For details see the [Barcodes HELP](#) system.

The dialog also contains OK and Cancel buttons. Pushing the OK button causes dBarcode to close the dialog and regenerate the barcode image for display in the Image area (or report an error if the image cannot be generated).

Pushing the Cancel button causes the dialog to be closed without effecting the currently displayed barcode image.

## Colour dialog

The Colour dialog contains a number of boxes showing the colours available on your Windows system, with the currently selected colour highlighted. Clicking on a colour box selects that colour.

The Colours dialog also contains OK and Cancel buttons, both of which close the dialog. If the OK button pushed is the currently selected colour will be used for the item nominated when the dialog was called (ie. either the foreground colour or the background colour) and the barcode image redrawn, while if Cancel is pushed the current barcode image display remains unchanged.



## Copy Options dialog

(32 bit versions only)

The Copy Options dialog contains check boxes which allow the user to select which formats are copied to the clipboard when a Copy command is executed.

The options are:

Standard metafile  
Enhanced metafile  
Bitmap

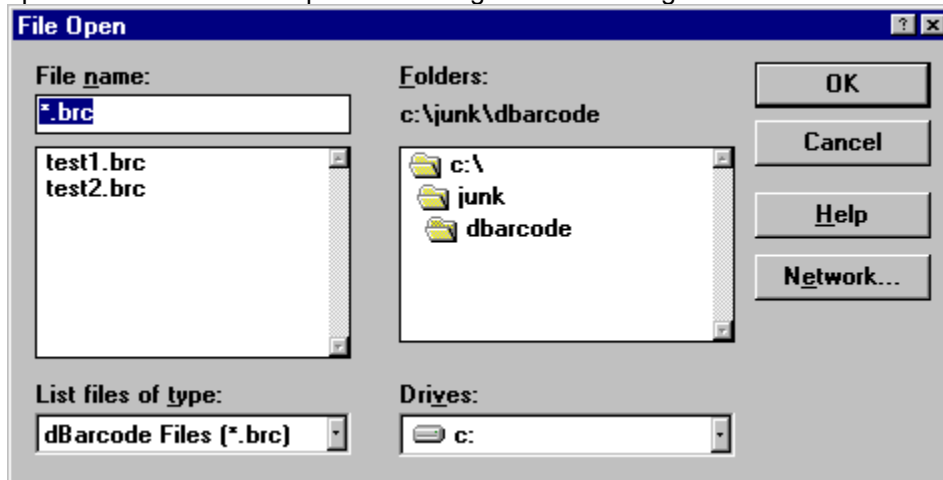
When an item is checked the corresponding format is copied to the clipboard on the next Copy command. If the dialog is closed with no items selected the standard metafile format is enabled. (See [Notes on Metafiles](#))

## File Open dialog

The Open file dialog box allows the user to select a configuration file for opening.

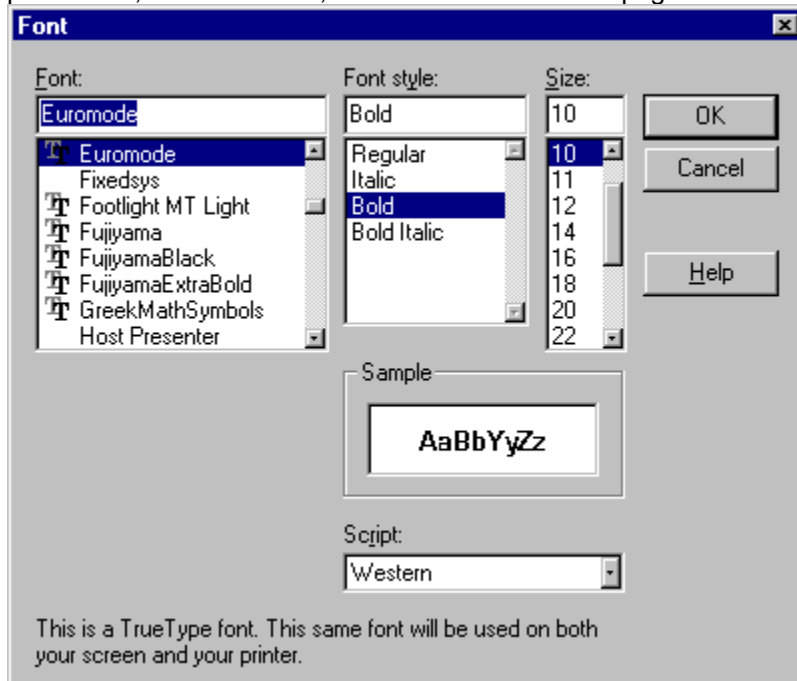
The dialog box shows the list of configuration filenames available in the current directory, using the default extension .BRC. An alternative drive and/or directory may be selected from the listboxes provided, and a suitable filename entered into the edit box.

Pushing the OK button causes the file to be loaded, while pushing the Cancel button cancels the operation and leaves the previous configuration unchanged.



## Font selection dialog

The Font selection dialog box allows the user to select the fontname, style (ie. regular, bold or italic) and size of the font which will be used to render the text. This selection is dependent on the current Windows printer and, in some cases, on the orientation of the page defined for that printer.



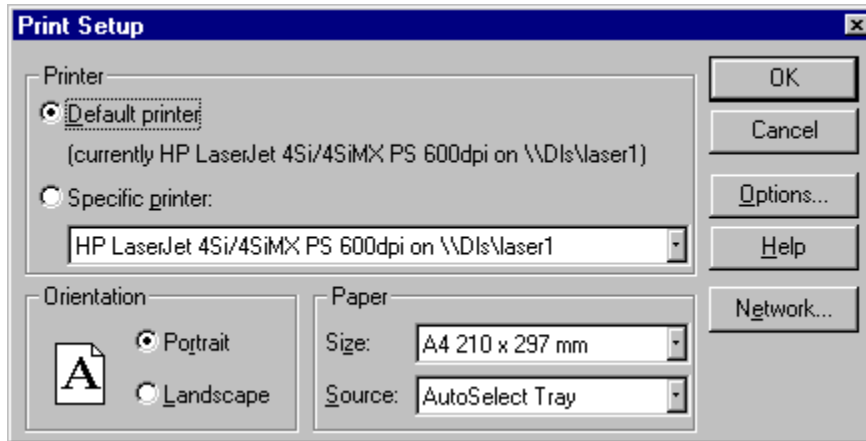
The Font selection dialog box contains the following items:

- Font listbox** Clicking on a fontname in the Font listbox selects the font for use in rendering the current component; this fontname is copied to the edit box above the list. It also causes the list of available font sizes in the size listbox to be updated. Note that only fonts available for the current printer are included in the Font listbox.
- Size listbox** Clicking on a font size in the Size listbox selects the size for use in rendering the current component; this size is copied to the edit box above the list. For TrueType, PostScript and other continuously scaleable fontsize printers a fixed number of options appear in the size listbox, although any integer point size in the range 4 - 144 points may be entered in the edit box and will be used for printing if the requested value is within the printer's capability.
- Style listbox** The font style box shows the font styles available for the currently selected font (including regular, italic, bold, and bold and italic). Clicking on one of these entries will select the style with which the font will be rendered on the screen image.
- Sample** The sample box shows a sample of the currently selected font and style.

The Font selection dialog box also contains OK and CANCEL buttons. When OK is selected any changes made within the dialog will be reflected in the Barcode image. If CANCEL is selected any such changes are discarded.



## Print Setup dialog



The print setup dialog allows the user to select the printer to be used for printing, and for deriving fonts used in barcode images.

The choice between the current default Windows printer and another selected printer may be made by pushing the appropriate radiobutton. If a selected printer is required the choice may be made from the drop down list of printer names.

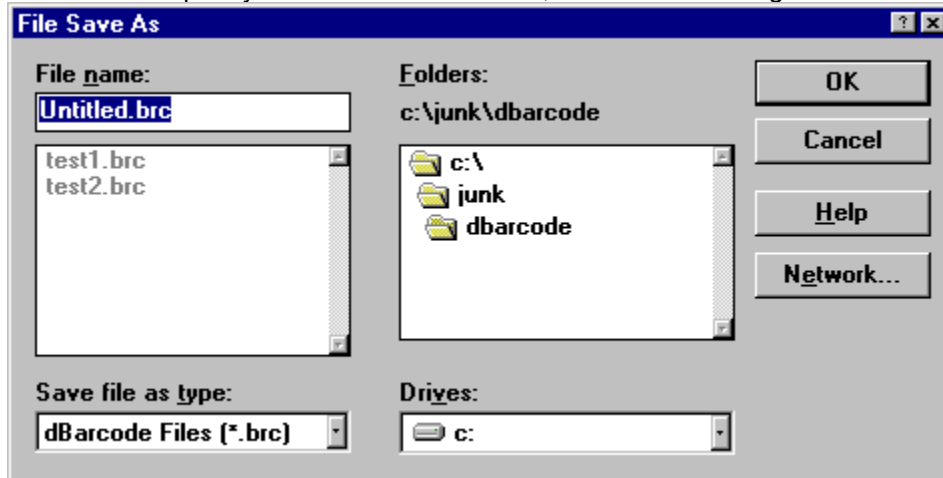
The selected printer's setup may also be checked and modified by pushing the Options button, which summons the Windows printer driver Setup dialog box.

## File Save dialog

The File Save dialog box, which allows the user to name a file for saving the current configuration or the current barcode file image.

The dialog box shows configuration or metafile files in the current directory, and contains listboxes which permit alternative drives, directories and file types to be selected.

Style (configuration) files are automatically saved with the default extension .BRC unless an alternative extension is explicitly included in the filename, while metafile images are saved with the extension .WMF.



Barcode images may be saved as bitmap images in either Windows bitmap files (with the extension BMP), standard PCX files (extension PCX) or in Encapsulated PostScript files (extension EPS), by selecting the required file type from the file type combo box.

# Embedded Objects

## Professional version only

dBarcode can be used as a stand-alone application as outlined in the introduction. However, dBarcode Pro is also an OLE-2 application which generates its barcode images as Objects, each one of which is known as a **dBarcode image**.

Note: Before dBarcode Pro is used as an OLE server for the first time, it is necessary to run the DBARCODE.EXE program as a stand-alone application so that dBarcode can register itself with your Windows's Registration database. If you ever need to reinstall Windows, or if you move the DBARCODE.EXE file to a different subdirectory you will need to run the DBARCODE.EXE program again to ensure that the registration database is updated.

## More:

[Inserting a dBarcode image into an OLE application](#)

[Standard Editing](#)

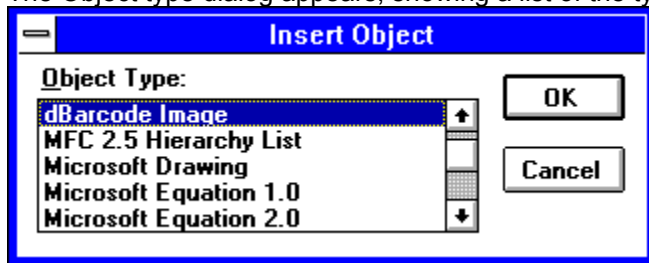
[In-Place editing](#)

[OLE Automation](#)

## Inserting a dBarcode image into an OLE application

To insert a dBarcode image into an OLE application simply position the cursor at the point where the barcode is required and then select that application's Insert Object command - which is usually located on the application's INSERT menu, although on older applications it may be on the FILE or EDIT menu. (see your application's manual for details).

The Object type dialog appears, showing a list of the types of object available for insertion.



Select dBarcode Image from the list of available type by clicking on it with the mouse pointer, and then push the OK button.

What happens next depends on whether your application supports Visual Editing (or In-Place editing) or not. Most applications do not support Visual editing, so we first discuss the more common standard editing, using the Write application as an example Client application.

## Standard Editing

So we continue where Insert object has been selected from Write's EDIT menu, and a dBarcode image has been selected from the Objects dialog.

The dBarcode window appears on screen, looking essentially the same as the stand-alone application. However, some of its menuitems are different from those described for the stand-alone application.

The File menu now contains the items:

- Update            update the dBarcode image in the client application
- Save Copy As    save a copy of the current configuration in a named file
- Load            Load a configuration from a named file
- Exit & Return    update the dBarcode image in the client application, close the dBarcode window and return to the client application



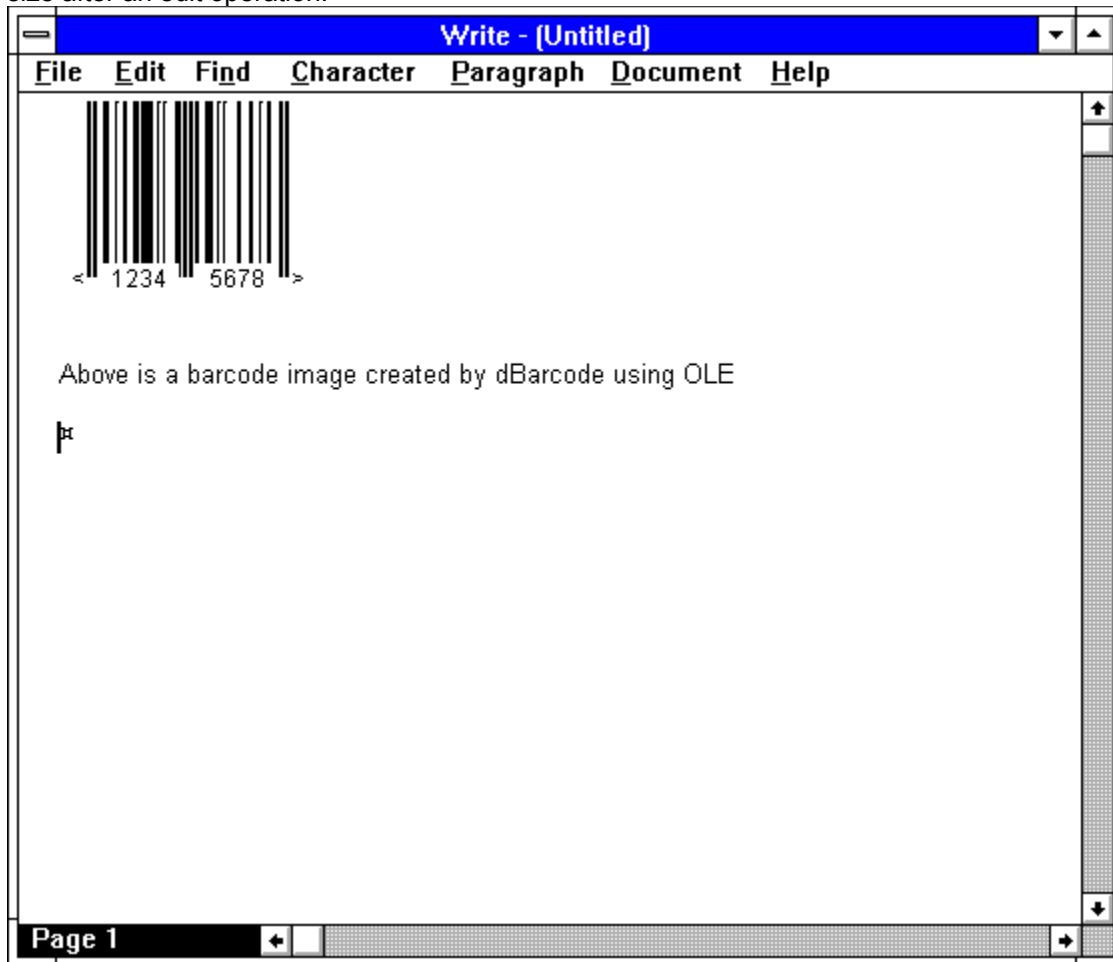
Initially the dBarcode window will show an image of the barcode specified as the default image (this may be changed using the Save Defaults command of the OPTIONS menu).

All the normal editing facilities are available in the dBarcode window, so the barcode may be adjusted as required.

When the image is as required the dBarcode window may be closed and control returned to Write by selecting Exit & return to Write from dBarcode's FILE menu.

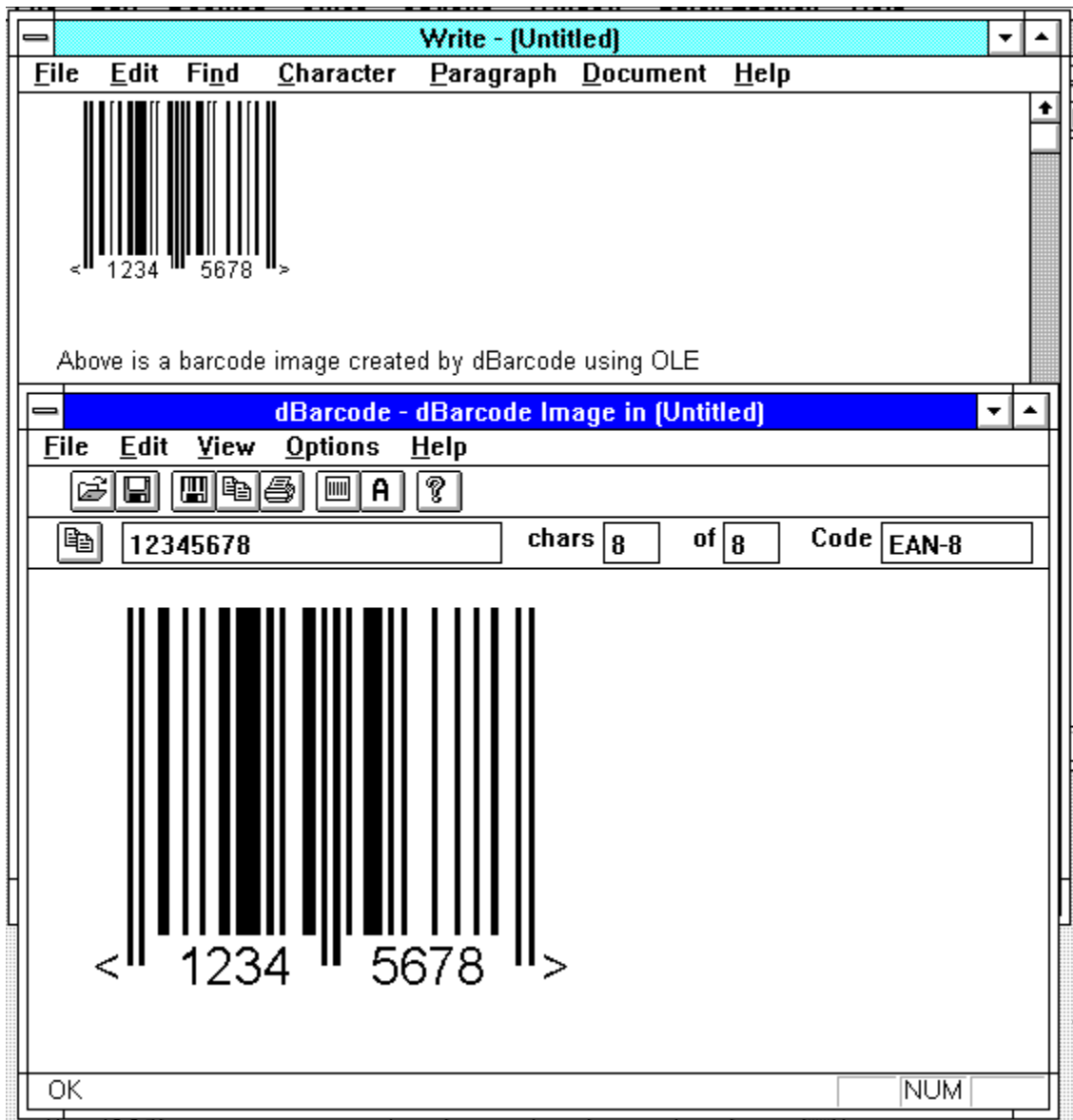
The dBarcode window closes and the Write window now shows the dBarcode image at the required point. The image may be moved around within the Write window using the normal Write controls.

Note that the image has been created at the size specified by dBarcode (in the Target Size boxes). The image may be resized (using Write's Size facilities) - but after resizing dBarcode will not reset the image size after an edit operation.



The dBarcode image within the client application may be edited at any time, either by double clicking on it, or by choosing **Edit dBarcode Image object** from the application's EDIT menu. The dBarcode window reappears showing the dBarcode image and ready for editing using the techniques discussed for the stand-alone application.

If a different barcode type is selected then the barcode image in the client application will be resized to suit the new image, but if image scaling has been carried out by the user using the client application's size or scale facilities, then dBarcode's image will remain scaled and will not be the size specified in dBarcode's Target Size boxes.



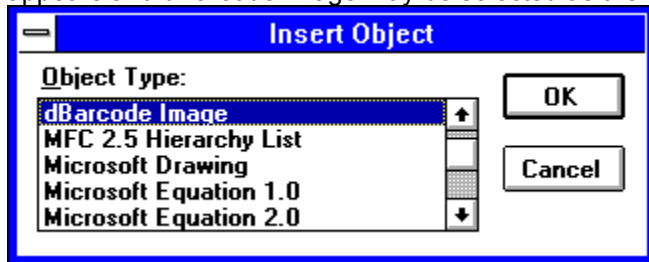
Again to complete the editing process and close dBarcode's window, choose Exit & Return to Write from dBarcode's FILE menu.

## In-Place editing

Some modern applications, such as Microsoft Word for Windows 6, support OLE In-place editing (also called Visual editing) in addition to the standard Embedded Object editing described above.

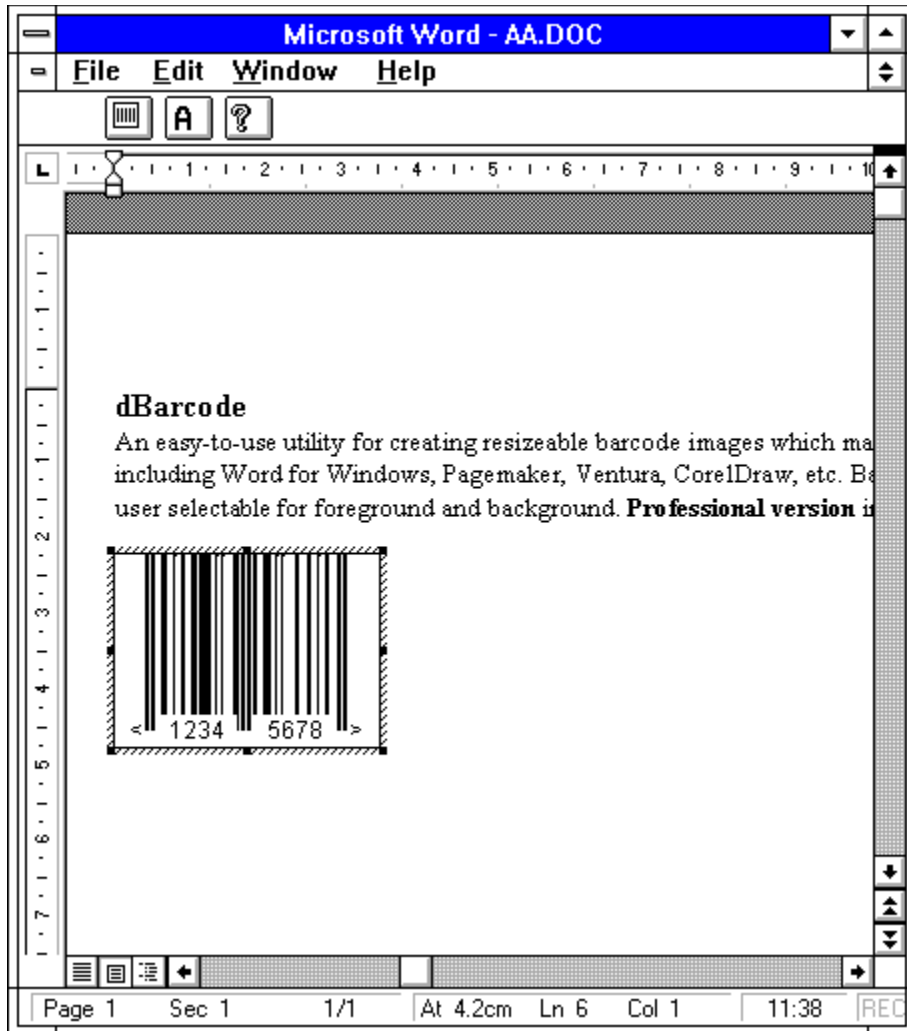
dBarcode image In-Place image creation and editing are described using Microsoft Word for Windows 6 as an example. For other applications see your application's documentation for details.

To create a dBarcode image within Word first position the cursor (insertion point) at the required position within the current Word document, then select Object from Word's INSERT menu. The Objects dialog appears and dBarcode Image may be selected as the appropriate object type.



An image of dBarcode's default barcode appears within the Word document at the insertion point and surrounded by an outline frame. At the same time Word's menu and Toolbar (if visible) change to reflect the editing facilities provided by dBarcode.



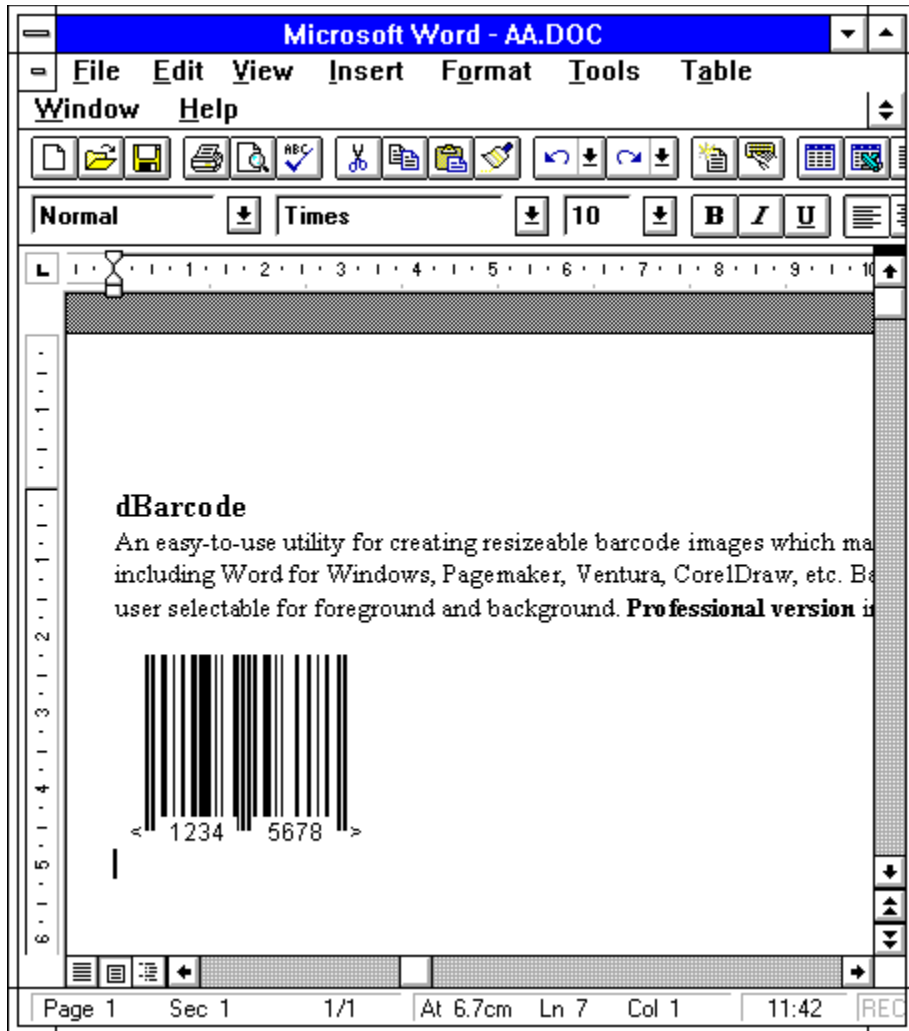


The actual menus displayed by a client application will depend on the application, but in all cases the EDIT menu should change to contain only the items:

- Copy - which copies the image to the clipboard
- Barcode Style - which summons the dBarcode Style dialog
- Barcode Font - which summons the Font properties dialog
- Load barcode - which summons dBarcode's File Open dialog to load a barcode configuration from a named file
- Save defaults - which saves the current configuration as the default configuration (used when a new image is created).

The Toolbar contains dBarcode's Code Style and Font buttons just as in the stand-alone application.

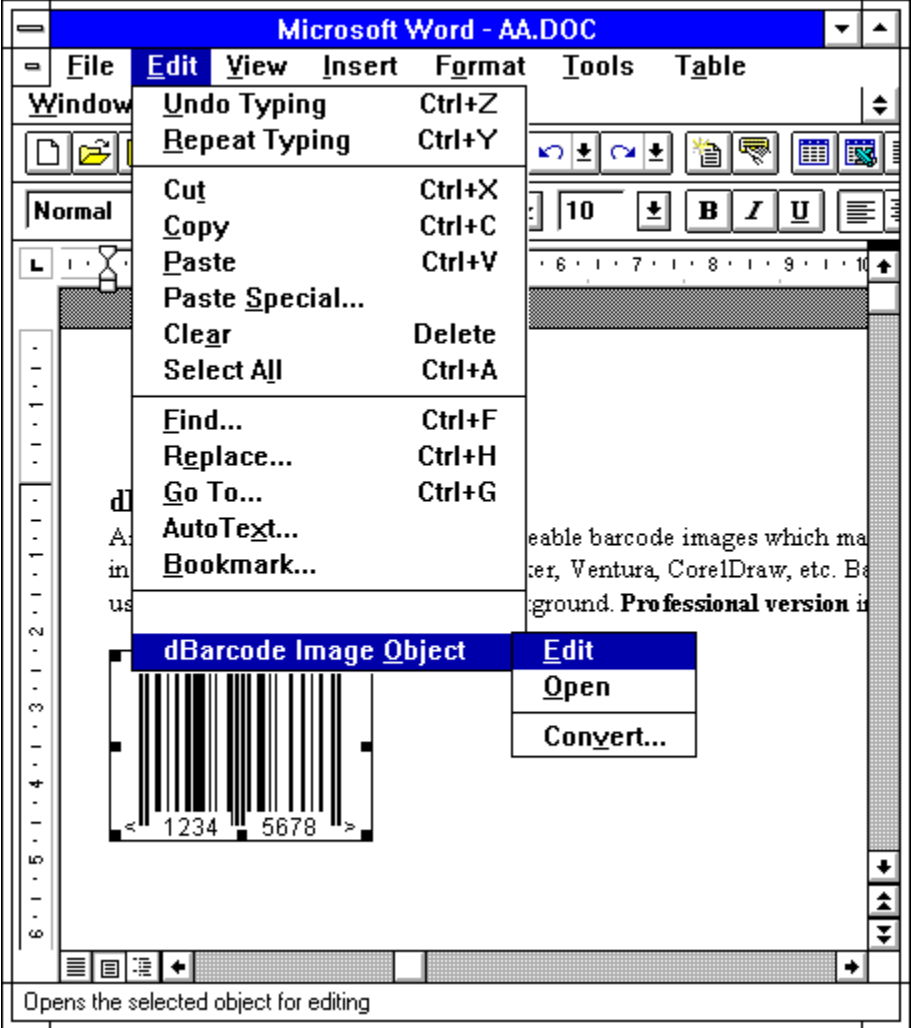
Once editing is completed simply clicking on the Word document anywhere outside the barcode image will return the standard Word display with the dBarcode image no longer surrounded by a frame.



Editing of the dBarcode image may be accomplished by *either* double clicking on the image - whereupon the Word menus and Toolbar are replaced by dBarcode's as described above, or by selecting dBarcode image Object from Word's EDIT menu.

Selecting dBarcode image Object from Word's EDIT menu causes the display of a sub-menu which contains the entries:

- Open**            Open the dBarcode window for editing the dBarcode image (see Standard Editing)
- Edit**            Change Word's menu to dBarcode's for In-Place editing
- Convert**        Convert the dBarcode image Object into an alternative object.



# OLE Automation

OLE Automation allows dBarcode Pro to be programmed from other applications, such as Microsoft Visual Basic 3. dBarcode provides several functions which may be called from other applications (one of which returns the barcode metafile image through the Windows clipboard), and a number of variables which may be set to specify properties of the barcode image, or retrieved to determine characteristics of the image.

The dBarcode functions provided for OLE automation are:

short dBarInit();

short dBarCopy();

short dBarMake()

short dBarMakeCode(codestring);

short dBarOpen(filename);

short dBarSave(filename);

short dBarDefs();

short dBarSaveImage(filename);

while the variables used for setting the properties of the barcode image are summarised in the table on the next page and described in detail below.

Most users will probably find it helpful to examine the sample programs provided on the distribution disk while reading the descriptions below.

## dBarcodes OLE variables

<b>type</b>	<b>name</b>	<b>purpose</b>
short	m_icode	code type identifier
short	m_ishowtext	include text flag
short	m_iautoparity	automatic parity digit flag
short	m_iorientation	image orientation indicator
short	m_ishowcheck	show check digit flag
short	m_iborderthick	image border thickness (points)
short	m_ilinereduce	bar thickness reduction (%)
short	m_ifontheight	text font height (points)
short	m_ifontbold	bold font flag
short	m_ifontitalic	italic font flag
short	m_ierror	Error code return
short	m_ixtra1/2	Additional property settings
short	m_ilength	image length
short	m_iheight	image height
short	m_irequired	characters required by code type
short	m_inominalsize	nominal size flag
short	m_imargins	margin indicator flag
short	m_ibearers	horizontal bearer bars flag
short	m_isspace	character spacing (% of length)
short	m_itransparent	transparent background flag

long	m_lforecolor	bar colour (RGB value)
long	m_lbackcolor	background colour (RGB colour)
CString	m_scode	code for conversion to barcode
CString	m_sfontname	name of font for text
CString	m_spattern	0/1 pattern returned

### **More:**

[dBarInit\(\)](#)

[dBarCopy\(\)](#)

[dBarMake\(\)](#)

[dBarMakeCode\(codestring\)](#)

[dBarOpen\(filename\)](#)

[dBarSave\(filename\)](#)

[dBarDefs\(\)](#)

[dBarSaveImage\(filename\)](#)

[OLE variables](#)

[A sample program](#)

## **dBarInit()**

Calling dBarInit() causes dBarcode to initialise the OLE variables with default values. This function should be called before any of the other control functions are called unless you are providing values for all of the OLE variables before calling the functions.

The default values for the OLE variables are maintained in the DBARCODE.INI file located in the Windows directory. This file is created or updated by either:

- a) running dBarcode in Stand-alone mode and choosing Save Defaults from the OPTIONS menu,
- or b) setting the values of all the OLE variables, calling the function dBarMake(), and then calling the dBarDefs() function (see below).

This function always returns 0.

## **dBarCopy()**

This function causes dBarcode to create the barcode image specified by the current setting of the OLE variables and pass the image to the Windows clipboard.

If an error is encountered while generating the image the variable `m_ierror` is set to a value greater than zero.

The OLE variable `m_spattern` is filled by this function.

The size of the metafile picture created on the clipboard will be that specified by the `m_ilength` and `m_iheight` variables, although these will themselves be set by dBarcode if the variable `m_inominalsize` is set to 1 and the barcode type has a nominal size.

The function returns 0 if an error is encountered, otherwise it returns a non-zero value.

## **dBarMake()**

This function causes dBarcode to create the barcode image specified by the current setting of the OLE variables. The image is not copied to the Windows clipboard by this command. dBarCopy must be used if the image is to be returned through the clipboard.

If an error is encountered while generating the image the variable m\_ierror is set to a value greater than zero.

The OLE variable m\_spattern is filled by this function.

The function returns 0 if an error is encountered, otherwise it returns a non-zero value.



## **dBarMakeCode(codestring)**

This function causes dBarcode to create the barcode image specified by the current setting of the OLE variables, but with the code for conversion being passed as the parameter *codestring*. The image is not copied to the Windows clipboard by this command. dBarCopy must be used if the image is to be returned through the clipboard.

Note that in this case any previous setting of the OLE variable *m\_scode* is lost. In all other respects this function behaves as dBarMake().

If an error is encountered while generating the image the variable *m\_ierror* is set to a value greater than zero.

The OLE variable *m\_spattern* is filled by this function.

The function returns 0 if an error is encountered, otherwise it returns a non-zero value.

## **dBarOpen(filename)**

This function when called with the parameter *filename* containing the full pathname of a dBarcode configuration file, loads the specified configuration file and generates the corresponding barcode image.

If the specified file does not exist this function returns 0 and does not change the current barcode image.

If the function is successful it returns a non-zero value.

This function provides the programming equivalent of the File OPEN command, complete with filename selection.

## **dBarSave(filename)**

This function when called with the parameter *filename* containing the full pathname of a dBarcode configuration file (complete with .BRC extension), saves the current barcode image configuration as a dBarcode configuration file.

If an error is encountered while attempting to save the file this function returns 0.

If the function is successful it returns a non-zero value.

This function provides the programming equivalent of the File SAVE AS command, complete with filename selection.

## **dBarDefs()**

Calling dBarDefs causes the default values of the OLE variables (which are stored in DBARCODE.INI) to be set equal to the current settings of the OLE variables - ie those use to generate the current image.

This function always returns 0.

## **dBarSaveImage(filename)**

This function when called with the parameter *filename* containing the full pathname of a file (complete with .WMF extension), saves the current barcode image as a Windows metafile file.

If an error is encountered while attempting to save the file this function returns 0.

If the function is successful it returns a non-zero value.

This function provides the programming equivalent of the File SAVE IMAGE command, complete with filename selection.

## OLE variables

### short m\_iautoparity;

Default value 0 This variable specifies whether a checkdigit should be automatically generated when a call to dBarMake generates the barcode image.

Allowed values:

0 - no automatic calculation

1 - checkdigit is automatically calculated

Note that for barcode types which do not have check digits this parameter is ignored.

### short m\_ibearers;

Default: 0 if =1 this causes the barcode image to be created with bearer bars (horizontal lines above and below ITF barcodes) included.  
If = 0 bearer bars are not included.

Allowed values: 0 or 1

### short m\_iborderthick;

Default value 0 This variable specifies the thickness (in points) of any border required around the barcode image. Any border is created using the colour specified as the background colour.

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

### short m\_icode;

Default value 8 The value of this variable specifies the code # as defined in the barcode types table.

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 Pro. The barcodes help file will be updated along with the DLL as new barcode types are added. For this reason the barcodes help file may be more up to date than the appendix provided in printed documentation.

### short m\_ierror;

Returns a value representing the error code if a valid barcode image cannot be created on a call to BarCodem. The error codes are shown below.

<i>code</i>	<i>meaning</i>
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
7	font error
8	unable to create metafile
9	no code provided

## **short m\_iextra1 and m\_iextra2**

Default 0

Allowed values 0 & 1. These values permit the setting of specific options for some barcode types. See the barcodes HELP file for details.

## **short m\_ifontbold;**

Default: 0 if = 1, this variable causes the font used for the text version of the code to be bold.  
If =0 the font is normal weight.

Allowed values: 0 or 1

## **short m\_ifontheight;**

Default: 10 the font height (in points) used for the text version of the code (if any) written under the barcode image.

Allowed values: 4 - 144 (points)

## **short m\_ifontitalic;**

Default: 0 if = 1, this variable causes the font used for the text version of the code to be italic.  
If =0 the font is normal.

Allowed values: 0 or 1

## **short m\_iheight;**

Default: 2500 Specifies the target height (in Himetric units) of the barcode image to be created. The target height includes any margins required by the barcode specification.

After a call to dBarMake in which m\_nominalsize was set to 1, this variable contains the nominal size height specified for the barcode type for those barcodes which have a nominal size. For barcodes which do not have a defined nominal size, m\_iheight remains unchanged after a call to dBarMake().

Allowed values: 0 - 32000 (units 0.01 mm)

## **short m\_ilength;**

Default: 3500 Specifies the target length (in Himetric units) of the barcode image to be created. The target length includes any margins required by the barcode specification.

After a call to dBarMake in which m\_nominalsize was set to 1, this variable contains the nominal size length specified for the barcode type for those barcodes which have a nominal size. For barcodes which do not have a defined nominal size, m\_ilength remains unchanged after a call to dBarMake().

Allowed values: 0 - 32000 (units 0.01 mm)

## **short m\_ilinereduce;**

Default: 0 A percentage reduction applied to the line thickness of each line drawn on the barcode image. This property may be used to compensate for ink spreading during wet-ink printing.

Allowed values: 0 - 50 (%)

## **short m\_imargins;**

Default: 0 if =1 this causes the barcode image to be created with margin indicators (< and > symbols for EAN codes, and vertical lines at the ends of ITF barcodes) included.  
If = 0 bearer bars are not included.

Allowed values: 0 or 1

### **short m\_inominalsize;**

Default: 0      if =1 this causes the barcode image to be created with dimensions of the nominal size specified for the codetype. If = 0 the size specified by the m\_ilenh and m\_iheight variables is used.

Allowed values: 0 or 1

### **short m\_iorientation;**

Default value 0      This variable specifies whether the barcode image should be generated normally or with clockwise or anticlockwise rotation through 90 degrees.

Allowed values

0 - Normal orientation

1 - clockwise rotation by 90 degrees

2 anti-clockwise rotation by 90 degrees

Note that only TrueType or compatible fonts may be rotated

### **short m\_irequired;**

Return value:      For the current barcode type (ie *m\_icode* in a call to dBarMake), this variable returns the number of characters required for a valid code, or 0 for coding schemes which do not specify a number of characters.

### **short m\_ishowcheck;**

Default value 0      This variable specifies whether any checkdigit should be included with the code text under the barcode image.

Allowed values:

0 - checkdigit is not included

1 - checkdigit is included

Note that for barcode types which do not have check digits this parameter is ignored. It is also ignored if text inclusion has not been requested.

### **short m\_ishowtext;**

Default value 0      This variable specifies whether a text version of the barcode should appear (usually underneath the bar image) when a call to dBarMake generates an image.

Allowed values:

0 - no text appears

1 - text appears, using the font specified by the m\_ifontheight and m\_sfontname variables.

### **short m\_isspaceing**

Default: 0      This variable specifies how much the text under the barcode is spread out along the length of the barcode. If the value is 0 then the text is created with its normal character spacing.

Allowed values:



0 - normal Windows character spacing used

10 - 100% - characters are spaced out to occupy the specified percentage of the barcode length

### **short m\_itransparent;**

Default: 0            if = 1, this variable causes the background of the barcode image to be transparent, and any background colour is ignored. If =0 the background is opaque.

Allowed values:    0 or 1

### **long m\_lforecolor;**

Default: 0            Set the colour of the image foreground, i.e. the bars and text colour.

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

### **long m\_lbackcolor;**

Default: &HFFFFFF&    Sets the colour of the image background.

Allowed values:    0 (black), to &HFFFFFF& (white)

### **CString m\_scode;**

Default: "12345"    This variable must contain the code string from which the barcode image is to be created before a call to dBarMake.

Note that in barcodes the difference between upper and lower case letters is significant. Few coding schemes support lower case letters.

### **CString m\_sfontname;**

Default: "Arial"    This variable contains a string which specifies the name of the font which will be used to create any text appearing under the barcode image.

An exact match is required between the string provided and a font existing on the Windows installation. For this reason it is advisable to obtain a user selected fontname from a source which will reliably provide a valid name - such as the Common Dialog controls of Visual Basic.

### **CString m\_spattern;**

Returns a string of 0s and 1s representing the bars (1s) and spaces (0s) in the barcode after each call to dBarMake. Users may use this string to draw their own rectangles.

## A sample program

The annotated part-program listing below illustrates the creation of a Code 39 barcode for the code 123456 using OLE automation from the Visual Basic (version 3) container. The program assumes that a form containing an Image (Image1) has been created for display of the barcode.

```
Set a = CreateObject("dBarcode.Image")      'start dBarcode
a.dBarInit                                  'initialise variables
a.icode = 8                                 'set code 39
a.ishowtext = 1                             'set text on
a.lforecolor = 0                            'bar colour black
a.lbackcolor = &HFFFFFF                     'background white
a.ifontheight = 10                          'font 10 point
a.sfontname = "Arial"                       'font Arial
a.ilength = 3500                             'length 35 mm
a.iheight = 2500                             'height 25 mm
a.scode = "123456"                          'code 123456
a.dBarCopy                                  'make image
i = a.ierror                                 'retrieve error code
if (i>0) then Print "Error";i                'check if non-zero
Image1.Picture = Clipboard.GetData(3)        'copy picture to Image1
```

