This manual was produced using ComponentOne Doc-To-Help.™
## Contents

### dBBarcode-2D

1. Introduction ................................................................. 1
2. Installation ........................................................................ 1
3. Starting dBBarcode-2D ...................................................... 1
4. Using dBBarcode-2D .......................................................... 2
   - Creating a barcode image ................................................. 2
   - Copying a barcode image to the Windows Clipboard ....... 2
   - Saving a barcode image as a file .................................... 3
   - Notes on Metafiles .......................................................... 3
   - Bitmap images ................................................................ 3

### Reference

1. The dBBarcode Window .................................................... 5
2. Menu bar .......................................................................... 5
3. The Tool bar ..................................................................... 5
4. Image area ......................................................................... 6
5. Status bar .......................................................................... 6
6. Menu Reference .............................................................. 7
   - File Menu ...................................................................... 7
   - Edit Menu ..................................................................... 7
   - View Menu ..................................................................... 7
   - Options Menu ............................................................. 8
   - Help Menu .................................................................... 8
7. Command Reference ........................................................... 8
8. Dialogs .............................................................................. 11
   - Code Properties dialog .................................................. 11
   - Colour dialog ................................................................ 13
   - File Open dialog ........................................................... 14
   - Graphic Options dialog .................................................. 15
   - Print Setup dialog .......................................................... 15
   - File Save dialog ............................................................ 16
   - Text Editor dialog .......................................................... 16

### OLE

1. Embedded objects ............................................................ 19
   - Inserting a dBBarcode image into an OLE application .... 19
   - Standard Editing ........................................................... 19
   - In-Place editing ............................................................ 20
2. OLE Automation ............................................................... 22
   - OLE methods ............................................................... 23
   - OLE properties ............................................................ 24
   - A sample program .......................................................... 28

### Index

31
dBarcode-2D

Introduction

dBarcode-2D is a Windows utility that enables two dimensional barcode images to be created and inserted into other Windows applications.

dBarcode images are normally created as Windows Metafiles. These are continuously scaleable and maintain a high quality at any size. However, dBarcode can also create bitmaps and other graphic formats and save them such as BMP, EPS, PNG, TIF, etc.

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 enabled applications.

Please note that before dBarcode-2D 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.

Installation

If you obtained dBarcode on CD then follow the instruction which came with the CD, or run Setup from the CD and follow the on-screen instructions.

If you obtained dBarcode electronically, then run the self-extracting executable file. This will expand the file contents to a temporary directory and run its own setup program from there. Again, just follow the on-screen instructions.

In either case be sure to read the ReadMe text file that appears on the Start button shortcut to dBarcode after installation.

Starting dBarcode-2D

Click on the Start button and select Programs from the menu presented. Open the program group into which you installed dBarcode-2D, and then select the dBarcode-2D program.

If you will be using dBarcode frequently you may wish to put a shortcut on your desktop. To do this right-click on the desktop and select New, Shortcut from the popup menu. A dialog box appears, asking for the command line to start the program. Push the Browse button to find the file DBAR2D.EXE that will be in the installation directory. Follow the
remaining instructions in the dialog to complete the process. You can then start dBarcode-2D by clicking on the shortcut on your desktop.

If you are using the dBarcode Suite then the dBarcode-2D component may be started from the dBarcode Suite Control Panel by pushing the “Create 2D Barcodes” button.

---

### Using d-Barcode-2D

#### Creating a barcode image

You can create a barcode image either within the dBarcode-2D application or by using the Insert Object feature of many other applications – such as Microsoft Word. Here we examine making an image in the dBarcode-2D application. For information on creating images within other applications please refer to the OLE section.

The procedure for creating a barcode image is simple:

1. Select the Properties item on the Edit menu or on the control bar to display the Properties dialog

2. Choose the type of barcode from the Barcode Type combo box.

3. Check that the Size information is as required and that the Mode option is set to the required value.

4. Close the Properties dialog by pushing the OK button.

5. Select the Text item from the Edit menu to display the text editor, and enter the text you wish to use to form the barcode symbol. Clicking in the text editor window with the RIGHT mouse button will display an editing menu that allows text on the clipboard to be pasted in.

6. Close the Text editor by pushing the OK button.

7. The barcode image will be displayed if the text 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 text contains characters that are not allowed in the mode or barcode type you have selected. The Status bar will show the reason why the image has not been created. Check the barcode type and mode in the Properties dialog 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 copy the barcode image to the clipboard choose Copy from the Edit menu or push the copy button on the control bar.

**Barcode types**

Details of the types of barcodes supported are contained in the Barcodes-2D HELP system that accompanies dBarcode-2D. 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-2D 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 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.

**Saving a barcode images as bitmap, PCX or other graphic files.**

Barcode images may be saved as bitmap images in a variety of graphics formats: either Windows bitmap files (with the extension BMP), standard PCX files (extension PCX), Portable Network Graphics (extension PNG), etc, 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).

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.

Some older Windows applications do not correctly interpret the size information within Windows metafiles, and when an image is pasted from the clipboard it may appear at the wrong size.

**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 specified by the iPixelGain property – which is the value specified as the Target resolution in the Graphics option dialog.

The user is recommended to use a target resolution equal to the resolution of the intended printer. ALL barcode images should be tested in their final form before extensive print runs are undertaken.
The dBarcode Window

The dBarcode window contains five areas of importance:
- The Menu bar
- The Tool bar
- The Image area
- The Status 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, or the image properties to be edited
  - View: Display or Hide the Tool and Status bars
  - Help: Provides access to the HELP system

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

The Tool bar

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

From left to right the icons are equivalent to:
Image area

The Image area is that portion of the dB Bar code window in which the barcode image is displayed. The Image area is resized automatically as the dB Bar code window is resized, and cannot otherwise be altered. The image area will be displayed using a size which is the best fit within the dB Bar code window, while maintaining the bar height to width ratio specified in the Code Property dialog.

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) that 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 that is generated when dB Bar code 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:

Error messages

<table>
<thead>
<tr>
<th>Code</th>
<th>message</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK (i.e. No error)</td>
</tr>
<tr>
<td>1</td>
<td>Wrong code length</td>
</tr>
<tr>
<td>2</td>
<td>Invalid code type</td>
</tr>
<tr>
<td>3</td>
<td>Invalid parameter</td>
</tr>
<tr>
<td>4</td>
<td>Illegal character in code</td>
</tr>
<tr>
<td>5</td>
<td>Error in embedded code</td>
</tr>
</tbody>
</table>
6  Generated line width less than 1 unit
7  not used
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 dBBarcode's window.

---

**Menu Reference**

**File Menu**

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

- **New**  Start a new file with default settings
- **Open**  Open a named configuration file.
- **Save**  Save the current configuration.
- **Save As**  Save the current configuration as a named file.
- **Save Image**  Save the current image as a WMF or other file.
- **Print**  Print the current image.
- **Print Preview**  Display a WYSIWYG view of what would be printed by a Print command
- **Print Setup**  Summons the Windows printer setup dialog.
- **Exit**  Close dBBarcode.

**Edit Menu**

The Edit menu contains a COPY command that enables you to copy the current Barcode image to the Windows Clipboard, and commands for editing the text of the barcode and the barcode's properties.

- **Copy**  Copy Image to clipboard
- **Test**  Edit the text for the barcode
- **Properties**  Edit the properties for the barcode image
- **Copy Options**  Allows selection of image format copied to clipboard during a copy operation,

**View Menu**

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

The menu also contains a Refresh command that may be used for refreshing the window display if this becomes corrupted.
Options Menu

The Options menu contains the following items:

Graphic Options: which causes the display of the Graphic Options dialog for setting the default graphic image type and properties.

Save Defaults: which causes the current barcode settings to be saved as the default values. These values will be used whenever dBarcode is started in future.

Help Menu

The Help menu contains commands for accessing dBarcode’s HELP system and its About box. The latter shows the version number of the program.

Command Reference

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 .D2C.

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. The extension used determines the type of image saved.

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

Configuration files are automatically saved with the extension .D2C 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) or in a variety of other file formats 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 file type for the image may be selected from the drop-down list box of file types. Note that Windows Metafiles maintain the highest quality of image, and that this is the recommended format for saving barcodes. WMF files may be loaded directly into applications such as Word for Windows.

**File Batch Images Command**

Both the 1D and 2D versions of dBarcode provide the facility to create multiple image files based on data provided in a text file. This can be particularly useful if, say, hundreds of PNG files need to be created for a web site.

The operation is fairly straightforward:

1. Using Notepad (or any other word processor which can save in ASCII text format) create a index file which contains the data text which will be converted into the barcode. Type the data for each barcode on a separate line – the data items may NOT contain carriage returns. Save this index file into a convenient directory – this is the directory into which the image files will be placed.

2. Optionally each line of data may be followed by a tab and then the name which will be used to name the file which will be created for the barcode image. If the filename includes an extension (eg. .wmf) then that extension will be used to determine the type of image created. If the filename does not include an extension, then the default graphics type (set in the module's Graphic Options) will be used and the appropriate extension added. If no filename (and no tab) is included on the line, then the file is created with a name equal to the data and an extension taken from the Graphic Options default graphic type.

3. From the module's File menu select Batch Images. The Open Batch File dialog is displayed. Navigate to the directory containing your index file, select that file, and push the Open button.

That's all there is to it. The dBarcode-2D module will create the images in the same directory as the index file.

We are not aware of any limitation in the number of images which may be created (we have made 10,000), but users should be aware that some graphic types take up quite a lot of space – so that a hard disk can rapidly fill up in this automatic mode.

**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 that allows the target printer to be selected, and attributes of the printer (e.g. 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 menu item chosen the current barcode image area displayed in dBarcode is copied to the clipboard as a metafile picture.

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

Note that copying an image to the clipboard automatically erases the clipboard of any other information in picture format.

**Edit Text Command**

When selected the dBarcode Text Editor dialog box is displayed. This allows the user to edit the text that will be converted into the barcode image.

**Edit Code Properties Command**

When selected the dBarcode Code Properties 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 Properties dialog.

**Edit Copy Options**

When selected the Copy Options Dialog is displayed. This allows the formats that will be copied to the clipboard during a Copy command to be selected.

**Options Graphic Options Command**

When selected the Graphic Options dialog is displayed, allowing user to select the default graphic file type and allowed options for that type.

**Options Save defaults Command**

When selected the current configuration is saved within the Windows registry. This configuration is then the default configuration, which is loaded automatically and used each time dBarcode is started.

**View Toolbar command**

Choosing this menu item toggles the display of the dBarcode toolbar. The Toolbar display is enabled when this menu item is ticked.
**View Status bar command**

Choosing this menu item toggles the display of the dBaBarcode status bar. The status bar display is enabled when this menu item is ticked.

**View Refresh command**

Choosing this item causes dBaBarcode's window to be redrawn.

---

**Dialogs**

**Code Properties dialog**

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

![Code Properties dialog](image)

**Code type combo box**

This drop-down combo box lists the code types supported by dBaBarcode-2D. 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.

**Mode box**

The mode box allows the user to select the starting mode for barcode types that allow a choice of modes. If in doubt leave the mode set at the first entry in the drop-down list.
**Target Sizes box**

The Target Sizes box contains edit boxes that allow the user to enter width values (in units of hundredths of a millimetre) for the unit bar width (i.e. the thinnest bar) used to create the final image, and the height of each bar as a multiple of its width. Note that the metafile image produced by dBarcode may be resized independently of the target size, although the sizes stored within the metafile will be those derived from the bar widths defined in this dialog.

The Sizes area also contains a box that permits the size of any border required to be defined (in millimetres).

The Sizes area also contains an edit box that allows the percentage reduction of bar width to be specified. This is useful if the target printer causes the bars to be printed larger than expected, or if wet ink is used and spreads on drying.

The thickness of each line drawn on the barcode image is reduced by this percentage amount. If a negative value is used the widths of the bars are reduced by the percentage of the thinnest bar.

Allowed values: 0 - 50 (%)  100 - 150 (%)

Values greater than 100 may be used to increase bar widths. If a negative value is used the widths of the bars are increased by the percentage of the thinnest bar.

**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.

**Security level edit box**

Some barcode types permit a variety of security levels to be adopted when text is converted into a barcode image. This edit box allows the security level to be specified. PDF 417 supports security levels from 0 - 8, 0 is no additional security, while 8 is a large amount of security - but at the price of a significantly larger barcode.

For Aztec barcodes in Normal mode the security level may be set to 0 (for default security) or 1 - 99 to specify the percentage of the symbol to be devoted to error checking codes. For Compact and Full range modes this box accepts the number of “layers” to be used for the symbol (1-4 for Compact, 1-32 for Full range); for Runes the value is ignored.

This box will be greyed out for barcode types that do not support user-selected security levels.

**Aspect ratio edit box**

Some two dimensional barcode types require the user to choose the height to width ratio of the final image (in terms of rows and columns of bars). Codablock and PDF 417 allow the user to specify the height to width ratio, and RSS (Expanded stacked) allow the width of the symbol in segment pairs to be specified.

For Codablock F symbols the desired height to width ratio for the symbol. dBarcode will do its best to comply - but will determine the final result within the limits set by the symbology. If a negative value of -4 or more negative, then the absolute value of the nearest integer is used to specify the number of data characters per row of the symbol. (This is useful if it is desired to generate several symbols of the same width.

For RSS Expanded stacked symbols this value represents the width of the symbol in segment pairs. Fractional values are rounded to the nearest integer value.

For PDF417 symbols dBarcode allows the user to specify the desired height/width ratio of the target image by entering a positive value in the range 0.1 - 10.0, or to specify the desired ratio of the number of rows/number of codewords per row by entering a negative value.

Selecting Columns from the drop down list allows the user to specify the number of codeword columns to be used in the symbol (this value must be 3 or more).
Selecting Rows from the drop down list allows the user to specify the number of rows in the symbol (this value must be 3 or more).

In entering a value the user should bear in mind that, in general, PDF barcodes a best reproduced as tall thin images rather than short fat ones. Also the PDF417 standard requires a minimum of 3 rows.

This box will be greyed out for barcode types that do not support user-selected sizing.

Flag checkbox
The Flag checkbox may be checked to indicate that an Aztec symbol should be created assuming that the input text uses \texttt{<Esc>n} for FLG(n).

This box is not available for other codes.

Menu checkbox
The Menu checkbox may be checked to indicate that an Aztec Menu Symbol is to be created,

This box is not available for other codes.

Reverse Video checkbox
The Reverse Video checkbox may be checked to produce a reversed video Aztec symbol

This box is not available for other codes.

Transparent 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.

Background colour button
When pushed, the Colour 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.

Bars 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.

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 (i.e. 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.

![Color Dialog](image)

**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 list boxes 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.

![File Open Dialog](image)
Graphic Options dialog

The graphic options dialog allows the default graphic format use in image file saves to be pre-selected, and any associated options to be chosen.

For graphics format based on bitmaps, the Target resolution specifies the number of pixels per inch of the target printing device.

The Batch Separators section allows separator characters to be defined for use in the batch file generation of graphic images. (See File Batch Images)

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 radio button. 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 graphics files in the current directory, and contains list boxes which permit alternative drives, directories and file types to be selected.

Style (configuration) files are automatically saved with the default extension .D2C 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 PNG files (extension PNG), etc, by selecting the required file type from the file type combo box.

**Text Editor dialog**

The text editor dialog allows the text that will be converted into a barcode to be entered or edited. The usual editing shortcut keys apply.
Clicking with the right mouse button within the editor window causes a menu of editing commands to be displayed.
OLE

Embedded objects

dBarcode can be used as a stand-alone application as outlined in the introduction. However, dBarcode is also an OLE application which generates its barcode images as Objects, each one of which is know as a dBarcode2D image

Note: Before dBarcode is used as an OLE server for the first time, it is necessary to run the DBAR2D.EXE program as a stand-alone application so that dBarcode can register itself with you Window's Registration database. If you ever need to reinstall Windows, or if you move the DBAR2D.EXE file to a different sub-directory, you will need to run the DBAR2D.EXE program again to ensure that the registration database is updated.

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 dBarcode2D 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 newer applications do support Visual editing, but we first discuss the standard editing, using Write 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 menu items 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 Bar 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 Bar 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**

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

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

To create a dBarcode image within WordPad first position the cursor (insertion point) at the required position within the current WordPad document, then select Object from WordPad's INSERT menu. The Objects dialog appears and dBarcode2D 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 WordPad'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
- Text - which summons the dBarcode Text editor
- Properties - which summons the dBarcode properties dialog

The Toolbar contains dBarcode's Properties and Text editor buttons just as in the stand-alone application.

Once editing is completed simply clicking on the WordPad document anywhere outside the barcode image will return the standard WordPad 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 WordPad menus and Toolbar are replaced by DBarcode's as described above, or by selecting DBarcode image Object from WordPad's EDIT menu.

Selecting DBarcode2D image Object from WordPad's EDIT menu causes the display of a sub-menu that contains the entries:

- **Open**
  - Open the DBarcode-2D window for editing the DBarcode image (see Standard Editing)

- **Edit**
  - Change Word's menu to DBarcode's for In-Place editing

Some applications will also include:

- **Convert**
  - Convert the DBarcode image Object into an alternative object - such as a picture.

## OLE Automation

OLE Automation allows DBarcode to be programmed from other applications, such as Microsoft Visual Basic 4 (or later). DBarcode provides several functions that may be called from other applications (one of which returns the barcode metafile image through the Windows clipboard), and a number of variables that 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:

```c
short dBarInit();
short dBarCopy();
short dBarCopyBitmap();
short dBarMake();
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.

**OLE methods**

**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.

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_errorcode is set to a value greater than zero.

The OLE variable m_pattern is filled by this function.

The size of the metafile picture created on the clipboard will be returned in the m_ilength and m_iheight variables, based on the bar width and height values specified in m_xunit and m_yunit respectively.

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

**dBarCopyBitmap()**

This function behaves as dBarCopy, except that the image is placed on the clipboard in Bitmap format (CF_BITMAP).

The size of the bitmap image created is determined by the values of m_xunit and m_yunit, but in MILS rather than HIMETRIC (0.01mm) unit.

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_errorcode is set to a value greater than zero.

The OLE variable m_pattern 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 (complete with D2C extension) 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 .D2C 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 to be set equal to the current settings of the OLE variables - i.e. those used 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 extension) saves the current barcode image as a disk file.

If the extension .WMF is used then the image is saved as a Windows metafile. If .BMP is used then the image is saved as a Windows Bitmap. The supported types are: BMP, EPS, GIF, JPG, PCX, PNG and TIF.

[For the dBarcode-2D Developers Kit, the list of supported graphics types may be viewed using the drop-down list in the Save Image dialog of the dBarcode-2D application.]

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 properties**

Note: OLE published names do not include the m_ prefix.

**dBarcode's OLE variables**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>short</td>
<td>m_code</td>
<td>code type identifier</td>
</tr>
<tr>
<td>short</td>
<td>m_orientation</td>
<td>image orientation indicator</td>
</tr>
<tr>
<td>short</td>
<td>m_borderthickness</td>
<td>image border thickness (mm)</td>
</tr>
<tr>
<td>short</td>
<td>m_linereduction</td>
<td>bar thickness reduction (%)</td>
</tr>
<tr>
<td>short</td>
<td>m_errorcode</td>
<td>Error code return</td>
</tr>
<tr>
<td>short</td>
<td>m_iheight</td>
<td>image height</td>
</tr>
<tr>
<td>short</td>
<td>m_columns</td>
<td>Number of columns in the symbol</td>
</tr>
<tr>
<td>short</td>
<td>m_xunit</td>
<td>bar thickness in MILS</td>
</tr>
<tr>
<td>short</td>
<td>m_yunit</td>
<td>bar height in MILS</td>
</tr>
<tr>
<td>short</td>
<td>m_securitylevel</td>
<td>security level for Aztec, DataMatrix and PDF 417 codes</td>
</tr>
<tr>
<td>Type</td>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>short</td>
<td>m_startmode</td>
<td>The start mode for multi-mode types</td>
</tr>
<tr>
<td>short</td>
<td>m_flags</td>
<td>flags</td>
</tr>
<tr>
<td>long</td>
<td>m_forecolor</td>
<td>bar colour (RGB value)</td>
</tr>
<tr>
<td>long</td>
<td>m_backcolor</td>
<td>background colour (RGB colour)</td>
</tr>
<tr>
<td>CString</td>
<td>m_scode</td>
<td>code for conversion to barcode</td>
</tr>
<tr>
<td>CString</td>
<td>m_pattern</td>
<td>pattern string returned</td>
</tr>
</tbody>
</table>

**m_columns**

Type: short  
Default: 0  
Allowed values: 0 - 100.  
For Codablock F symbols the number of data characters per row of the symbol. (This is useful if it is desired to generate several symbols of the same width.  
For RSS Expanded stacked symbols the value represents the width of the symbol in segment pairs. Fractional values are rounded to the nearest integer.  
For PDF417 value of the m_aspectratio property specifies the number of codeword columns in the barcode symbol. A negative value of the m_aspectratio property specifies the number of rows in the symbol.  
Ignored for barcode types other than Codablock, PDF417 and RSS.

**m_backcolor**;

Type: long  
Default: &HFFFFFF&  
Sets the colour of the image background.  
Allowed values: 0 (black), to &HFFFFFF& (white)

**m_borderthickness**;

Type: short  
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 10

**m_code**;

Type: short  
Default value 0  
The value of this variable specifies the code # as defined below.  
Allowed values: 0 = Code 16k  
1 = Code 49  
2 = PDF 417  
3 = Aztec  
4 = DataMatrix  
5 = MaxiCode
The barcode types table is also shown in the barcodes help file supplied with the product library which accompanies dBarcode 2D. The barcodes help file will be updated along with the DLL as new barcode types are added.

**m_errorcode;**

Type: short  

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

<table>
<thead>
<tr>
<th>code</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No error</td>
</tr>
<tr>
<td>1</td>
<td>Wrong code length</td>
</tr>
<tr>
<td>2</td>
<td>Unrecognised code type</td>
</tr>
<tr>
<td>3</td>
<td>Wrong add-on code length</td>
</tr>
<tr>
<td>4</td>
<td>Illegal character in code</td>
</tr>
<tr>
<td>5</td>
<td>Error in embedded code</td>
</tr>
<tr>
<td>6</td>
<td>Generated line width less than 1 unit</td>
</tr>
<tr>
<td>7</td>
<td>Font error</td>
</tr>
<tr>
<td>8</td>
<td>Unable to create metafile</td>
</tr>
<tr>
<td>9</td>
<td>No code provided</td>
</tr>
</tbody>
</table>

**m_flags;**

Type: short  

Default: 0  

Set the properties that affect the image display.

Individual properties are combined by the OR operation. The only flags of significance in this version are:

- DL_FLAG_TRANSPARENT = 4 - which, when set, ensures that the barcode image background is transparent (i.e. no colour).
- DL_FLAG_EXTRA1 = 16 (see Barcodes-2D Help for details)
- DL_FLAG_EXTRA2 = 32 (see Barcodes Help for details)

For Aztec codes the following flags are defined:

- DL_AZFLAG = 16
- DL_AZMENU = 2
- DL_AZRVIDEO = 32

See barcodes Help for details.

**m_forecolor;**

Type: long  

Default: 0  

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

Allowed values: 0 (black) to &H00FFFFFF (white)
m_iheight;
Type: short
Returns the height of the barcode image after a call to dBarMake or dBarCopy.

m_ilength;
Type: short
Returns the width of the barcode image after a call to dBarMake or dBarCopy.

m_iPixelGain;
Type: short
Set the resolution (in dpi) to be used in the generation of bitmap images.

m_linereduction;
Type: short
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 (%)
Note that a positive value applies a reduction based on the size of a single unit bar, while a negative value applies a reduction that is proportional to the thickness of each bar. In general some experimentation may be necessary to determine the optimum value required for a specified printer.

m_orientation;
Type: short
Default value 0
This variable specifies whether the barcode image should be generated normally or with clockwise or anti-clockwise rotation through 90 degrees.

Allowed values
0 - Normal orientation
1 - clockwise rotation by 90 degrees
2 anti-clockwise rotation by 90 degrees
-1 inverted

Note that only TrueType or compatible fonts may be rotated.

m_pattern;
Type: CString
Returns a string of 0s, 1s and other numbers representing the bar and space patterns in the barcode after each call to dBarMake. Users may use this string to drawn their own rectangles. Each row of a stacked barcode is separated from the next by a newline character (\n or CHR$(10)).
m_scode;
Type: CString
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.

m_securitylevel;
Type: short
Default: 0  This variable must contain the security level code for PDF 417 and Aztec barcodes.
Allowed values
0-8 for PDF417
0-99 for Aztec Normal Mode
1-4 (layers) for Aztec Compact mode
1-32 (layers) for Aztec Full range mode
Note that the higher the value of the security level the larger will be the barcode. This value is ignored for Aztec Runes.

m_start mode
Type: short
Default: 0  The start mode for those barcode types that have multiple modes. See barcode Help for details.

m_xunit;
Type: short
Default: 20  The unit width of bars in MILS units (i.e. 0.001 inch)

m_yunit;
Type: short
Default: 200 - code 16k
200 - code 49
60 - PDF 417
The height of the bars in MILS units (i.e. 0.001 inch)

A sample program
The annotated part-program listing below illustrates the creation of a PDF 417 barcode for the code “DBARCODE 2D” using OLE automation from the Visual Basic (version 4) container. The program assumes that a form containing and Image (Image1) has been created for display of the barcode.

Set a = CreateObject("dBarcode2D.Image")  'start dBarcode
a.dBarInit  'initialise variables
a.code = 2  'set code to PDF 417
a.forecolor = 0  'bar colour black
a.backcolor = &HFFFFFF  'background white
a.xunit = 50         'bar width 0.5 mm
a.yunit = 500        'bar height 5 mm
a.scode = "DBARCODE 2D"   'code DBARCODE 2D
a.dBarCopy         'make image
i = a.errorcode    'retrieve error code
if (i>0) then Print "Error";i  'check if non-zero
Image1.Picture = Clipboard.GetData(3)  'copy picture to Image1
Index

A
A sample program 28
Aspect ratio edit box 12

B
Background colour button 13
Barcode types 2
Bars colour button 13
Bitmap images 3

C
Code Properties dialog 11
Code type combo box 11
Colour dialog 13
Command Reference 8
Copying a barcode image to the Windows Clipboard. 2
Creating a barcode image 2

D
dBarcode-2D 1
dBarcode's OLE variables 24
dBarCopy() 23
dBarCopyBitmap() 23
dBarDefs() 24
dBarInit() 23
dBarMake() 23
dBarOpen(filename) 23
dBarSave(filename) 24
dBarSaveImage(filename) 24
Dialogs 11

E
Edit Code Properties Command 10
Edit Copy Command 10
Edit Copy Options 10
Edit Menu 7
Edit Text Command 10

Embedded objects 19
Error messages 6

F
File Batch Images Command 9
File Exit Command 10
File Menu 7
File Open Command 8
File Open dialog 14
File Print command 9
File Print Preview command 9
File Printer Setup command 9
File Save As Command 8
File Save Command 8
File Save dialog 16
File Save Image Command 8
Flag checkbox 13

G
Graphic Options dialog 15

H
Help Menu 8

I
Image area 6
index file 9
In-Place editing 20
Inserting a dBarcode image into an OLE application 19
Installation 1
Introduction 1

M
m_backcolor; 25
m_borderthickness; 25
m_code; 25
m_columns 25
m_errorcode; 26
m_flags; 26
m_forecolor; 26
m_iheight; 27
m ilength; 27
m_issuer; 27
m_iPattern; 27
m_orientation; 27
m pattern; 27
m_scode; 28
m_securitylevel; 28
m_start mode 28
m_xunit; 28
m_yunit; 28
Menu bar 5
Menu checkbox 13
Menu Reference 7
Mode box 11

N
Notes on Metafiles 3

O
OLE 19
OLE Automation 22
OLE methods 23
OLE properties 24
Options Graphic Options Command 10
Options Menu 8
Options Save defaults Command 10
Orientation combo box 12

P
Print Setup dialog 15

R
Reference 5
Reverse Video checkbox 13

S
Saving a barcode image as a file 3
Saving a barcode images as bitmap, PCX or other
graphic files. 3
Security level edit box 12
Standard Editing 19
Starting dBarcode-2D 1
Status bar 6

T
Target Sizes box 12
Text Editor dialog 16
The dBarcode Window 5
The Tool bar 5
Transparent checkbox 13

U
Using d-Barcode-2D 2

V
View Menu 7
View Refresh command 11
View Status bar command 11