

Contents:

Contents:	1
License Agreement:	2
Introduction	3
Installing B-Coder On Your Computer	4
Getting Started	4
Using B-Coder With Other Windows Programs	5
The B-Coder Window	6
The Main Menu	8
File Menu Commands	9
The Raster Image Properties Dialog Box	12
Edit Menu Commands	14
Preferences Menu Commands	17
Symbology Menu Commands	20
Help Menu Commands	20
The B-Coder Toolbar	21
Bar Code Dimensions	22
Narrow To Wide Element Ratios	23
UPC and EAN Magnification Factors	23
PDF417 Bar Code Dimensions	24
Aztec Code and Data Matrix Bar Code Dimensions	25
How To Produce Readable Bar Codes	25
Bar Code Basics	26
How A Bar Code Reader Works	27
Special Considerations and Incompatibilities	28
Pasting Bar Codes Into 32 bit Programs (Word, Excel and Pagemaker 6.x, etc.)	28
Pasting Bar Codes Into Programs That Work With Bitmaps	28
Pasting Bar Codes Into A Spreadsheet	29
Printing Bar Codes On A Dot Matrix Printer	29
Bar Code Symbology Descriptions and Rules	30
CODE 39 (Normal and Full ASCII versions)	30
UPC-A, UPC-E, and UPC Supplementals	32
EAN-8 / EAN-13, BookLand and EAN Supplementals	33
CODABAR	33
INTERLEAVED 2 OF 5	34
DISCRETE 2 OF 5	34
CODE 93	35
MSI/Plessey	35
CODE 128	35
EAN/UCC 128	36
POSTNET	36
Postal FIM Patterns	36
PDF417	37
The PDF417 Text Editor	38
PDF417 Data Compaction Modes	39
PDF417 Error Detection and Correction	40
Truncated PDF417 Symbols	40
The B-Coder PDF417 Options Dialog Box.	41
BPO 4 State Code	43
RSS14	43
Aztec Code	44
Data Matrix	46
Understanding Dynamic Data Exchange	50
DDE Commands Supported By B-Coder	51
DDE Examples on the Internet	55
DDE Examples for Microsoft WORD	55
DDE Example for Microsoft Access	56
Using the LINKTEST Utility to test DDE Commands	58

License Agreement:

1. GRANT OF LICENSE

TAL Technologies, Inc. grants you the right to use one copy of the enclosed software program (the SOFTWARE) on a single terminal connected to a single computer (i.e., with a single CPU). You may not network the SOFTWARE or otherwise use it on more than one computer or computer terminal at the same time.

2. COPYRIGHT

The SOFTWARE is owned by TAL Technologies, Inc. and is protected by United States copyright laws and treaties. Therefore, you must treat the SOFTWARE like any other copyrighted material (e.g., a book or musical recording) except that you may either (a) make one copy of the SOFTWARE solely for backup or archival purposes, or (b) transfer the SOFTWARE to a single hard disk provided you keep the original solely for backup or archival purposes. You may not copy the users manual accompanying the SOFTWARE.

3. OTHER RESTRICTIONS

You may not rent or lease the SOFTWARE, but you may transfer the SOFTWARE and accompanying written materials on a permanent basis provided you retain no copies and the recipient agrees to the terms of this Agreement. You may not reverse engineer, decompile, or disassemble the SOFTWARE. If SOFTWARE is an update, any transfer must include the update and all prior versions.

4. DISCLAIMER

No Warranty of any kind on the SOFTWARE is expressed or implied.

In no event shall TAL Technologies, Inc. or its suppliers be liable for any damages whatsoever (including, without limitation, damages for loss of business profit, business interruption, loss of business information or other pecuniary loss) arising out of the use of or inability to use this product. Should you have any questions concerning this agreement, or wish to contact TAL Technologies, Inc. for any reason please write:

**TAL Technologies, Inc.
Customer Service Dept.
2027 Wallace Street
Philadelphia, PA 19130-3221 USA
Tel: (215)-763-5096 Fax: (215)-763-9711
e-mail: tal1@taltech.com**

Information in this document is subject to change without notice and does not represent a commitment on the part of TAL Technologies, Inc. The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of this agreement. It is against the law to copy the software on any medium except as specifically allowed in the license agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written consent of TAL Technologies, Inc.

© 1994-2003 TAL Technologies, Inc. All rights reserved. Printed in the United States of America

Introduction

Congratulations! You have purchased the most powerful, most versatile, and the easiest to use bar code software package available for MS Windows. B-Coder Professional has all the features necessary to produce professional quality bar codes for all types of applications including product packaging, document tracking, Postal bar coding and special purpose bar code labeling applications. Not only is B-Coder powerful, it is also extremely easy to use.

The philosophy behind B-Coder is that instead of forcing you to stay within the limits of a bar code labeling program, B-Coder acts as a bar code graphics tool for use with other Windows applications. With B-Coder you design the printed output or label layout with another program that you already know how to use like MS Word, Access, Pagemaker, Quark, WordPad, etc., you then use B-Coder to produce bar codes that are quickly and easily pasted into the other program.

Most other bar code software packages are either low quality bar code fonts or are complex label design programs that force you to use their own label design tools. B-Coder on the other hand creates high resolution bar code graphic images that you can incorporate into any Windows application program that you like!

B-Coder supports all commonly used bar code symbologies, it allows full selection for both the foreground and background colors, it supports symbol rotation in 90 degree increments and it allows complete font selection for the human readable text below a bar code symbol. More advanced features include message input from a disk file, automatic numeric sequencing, target printer selection, measurement unit selection and complete and accurate control over all bar code dimensions. B-Coder's intuitive warning messages make it virtually impossible to produce unreadable bar codes no matter what type of printer you are using.

The most powerful aspect of B-Coder is its ability to process commands sent to it from other applications. Anything that you can do from B-Coder's main menu can be done automatically under the control of other Windows programs. For example, you can easily create a fully automated bar code labeling system using a word processor like MS Word or WordPerfect by writing a simple macro in the word processor's macro language. For more information about using B-Coder's command language, see the sections: DDE Commands Supported By B-Coder (pg. 51) and the DDE Examples for Microsoft WORD (pg. 55).

Another advanced feature of B-Coder is its ability to produce two dimensional bar codes including PDF417, Aztec Code and Data Matrix. These new two dimensional symbologies are capable of encoding up to 3700 characters in a single bar code. For example, you could easily encode the entire Gettysburg address on a single bar code symbol measuring less than four square inches.

Installing B-Coder On Your Computer

B-Coder is provided with a "Setup" program that will install the B-Coder program files onto your computer. The setup program also optionally creates a B-Coder program folder in the Windows Start Menu with the B-Coder program icons in it.

To install B-Coder on your PC, place the B-Coder Setup disk in your CD ROM (or disk drive A or B if installing from floppy diskettes). Next, click the Start button and select **Run...** from the Start Menu. A dialog box will appear where you should enter the command: **D:SETUP** where *D* is the drive letter of the CD ROM or disk drive that you are installing from. After you press the **Enter** key the Setup program will prompt you for the name of a directory where you would like to install B-Coder and then it will copy all necessary files to the specified directory.

Getting Started

B-Coder is an extremely easy to use program that generates professional quality bar codes in all major symbologies and sizes and places them in the Windows clipboard so that they can be pasted directly into files or documents created with other Windows programs. Bar codes generated with B-Coder may be scaled to any size and can be used by any Windows program that supports pasting of Windows MetaFile graphics (i.e. MS Word, Windows Write, Excel, Word Perfect, Micrografx Draw, Pagemaker, Corel Draw, Visio, Quark etc.).

B-Coder can also be used to generate bar codes and save them to disk in practically every major graphics file format including EPS, TIF, GIF, JPG, PNG, WMF, BMP, TGA and PCX. This makes B-Coder ideal for all desktop publishing applications that require bar code images to be incorporated in a document, label or product package design.

B-Coder gives you the ability to quickly incorporate bar codes into any Windows application program. Instead of dealing with expensive and complex bar code label design programs, simply paste perfect bar codes directly into the programs that you already know how to use and print them on any printer supported by Windows (i.e. any standard printer).

B-Coder also supports a set of DDE commands that allow other Windows programs to completely control its operation. The DDE command language allows B-Coder to function as a bar code graphics "engine". Any Windows program that can issue DDE commands can use B-Coder to generate bar codes on demand. For example, you can use B-Coder with Word for Windows to create an extremely powerful bar code labeling system by writing one simple Word macro. (See DDE Examples for Microsoft WORD (pg. 55)) B-Coder even comes complete with a set of pre-written Word macros that you can easily install into your Word toolbar that make the task of adding bar codes to Word documents and labels a trivial operation.

Using B-Coder With Other Windows Programs

B-Coder was designed as a bar code generating tool to be used along with other Windows programs. The simplest way to use B-Coder is to run both B-Coder and the program that you want to paste bar codes into at the same time. When you need to insert a bar code into the other program, switch to B-Coder, enter the text that you want represented as a bar code in the B-Coder window and select **BUILD/COPY** from B-Coder's Edit menu. This will create your bar code and automatically copy it to the Windows clipboard. Finally switch back to your original program and select **PASTE** from the program's Edit menu. The bar code will appear in the program's window.

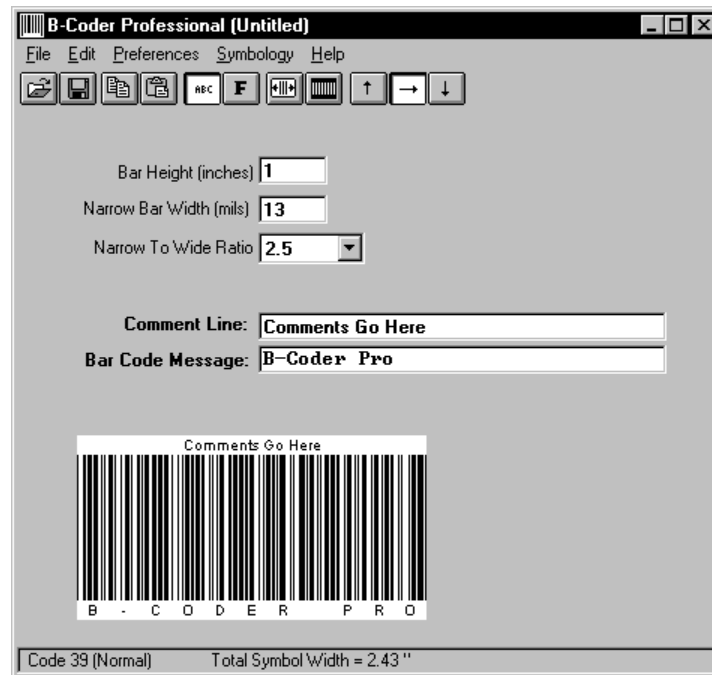
Note: Your monitor's resolution may not be high enough to display the bar code exactly as it will appear when you finally print the bar code on your printer. Also, some 32 bit Windows programs like Word and Pagemaker 6.x or above require you to use the **Paste Special** option in their Edit menu to paste a Metafile to avoid it being automatically converted to an Enhanced Metafile.

See also: Pasting Bar Codes Into 32 bit Programs (pg. 28)

The following samples were all pasted into Microsoft Word and were printed on an HP Laserjet 4 laser printer.



The B-Coder Window



The B-Coder window contains a main menu just below the title bar. A *tool bar* directly below the main menu allows quick access to some of the more useful items in the main menu. Below the tool bar, inside the B-Coder window, are several data input fields that allow control over the dimensions of your bar codes. These fields will differ depending on the features of each chosen bar code symbology. For details on all options available in the main menu see the section: The Main Menu (pg. 8). For details on the fields that control the dimensions of your bar codes refer to: Bar Code Dimensions (pg. 22) and (pg. 30).

The **Bar Code Message** text field in the center of B-Coder's window is where you enter the text that you want represented as a bar code. The **Comment Line** text field allows you to enter an additional comment line that is printed either above or below your bar code symbol.

At the bottom of B-Coder's window is a status area that displays the currently selected bar code symbology as well as the total overall width of each generated bar code symbol. The status area will also display any check digit values that B-Coder calculates for bar code symbols that include a check digit.

Note: If you press the Enter key while editing a bar code message, B-Coder will insert a carriage return linefeed pair in your message. These characters will appear as a musical note and a small rectangle in the message text. Unless you want a carriage return linefeed in your bar code, you should not press the Enter key when finished editing a bar code message.

Hint: Double clicking your *right* mouse button with the mouse cursor on any blank space in B-Coder's window will delete all text in the Bar Code Message text edit field and the Comment Line text edit field.

If the bar code symbology that you are using supports characters that cannot be typed in on your keyboard, an ASCII chart will also appear so that you can enter all characters in the symbology's character set. To enter characters in your bar code text using the ASCII chart, move the text cursor in the bar code text edit box to the point where you want to insert a specific character and then select the character by double clicking your mouse on it in the ASCII chart. The character will automatically appear in the bar code text edit

control. You may also select characters from the ASCII chart using the arrow keys or scroll bars to scroll the chart and then pressing the Enter key to select a hi-lighted character.

ASCII	Char	Control
0		NUL
1	☺	SOH
2	☻	STX
3	♥	ETX
4	♦	EOT
5	♣	ENQ
6	♠	ACK

Note: The Code 128 and EAN/UCC 128 symbologies supports not only the full 128 character ASCII character set, but they also support special *Function Codes*. These function codes can be found at the bottom of the ASCII chart when either of these symbologies is selected.
See Also: Code 128 in the section: (pg. 30)

The Main Menu



B-Coder's main menu contains the following sub menus: **File**, **Edit**, **Preferences**, **Symbology** and **Help**.

The **File** menu contains commands for performing disk file related operations like loading configuration files, saving bar code images to a disk file, selection of the target printer that will be used to print your bar codes and the Exit command.

The **Edit** menu contains commands for building and copying a bar code symbol, pasting text into B-Coder from other programs, editing of incremental numbering parameters and for scrolling forward and backward through a sequence of bar code messages.

The **Preferences** menu contains commands that allow you to customize the way that B-Coder works including such features as font and color selection, controlling symbology specific options, changing the orientation of bar code symbols, selection of measurement units, etc..

The **Symbology** menu allows you to select the type of bar code to produce. (The bar code symbology).

The **Help** menu provides quick access to B-Coder's on line help system. B-Coder also provides complete context sensitive help. If you have difficulty understanding any of the features available in B-Coder, you can get instant on-line help by pressing function key F1.

File Menu Commands

File	
New	Ctrl+N
Open Configuration...	Ctrl+O
Save Configuration	
Save Configuration As...	
Read Message From File...	
Open Message File...	
Close Message File	
Target Printer...	
Printer Setup...	
Print...	Ctrl+P
Save Bar Code As...	Ctrl+S
Raster Image Properties...	
Exit	

The first four options in the File menu: **New**, **Open Configuration**, **Save Configuration**, and **Save Configuration As...** allow you to create, open and save B-Coder configuration files. When you set up B-Coder by selecting all the various parameters necessary to produce the correct bar code symbols for your particular application, you can save your settings in a "configuration file" so that when you exit B-Coder, you can open your configuration and thus start up where you left off the next time you run B-Coder.

NOTE: You can automatically load a configuration file when you start up B-Coder by placing the name of a configuration file on the command line used to invoke B-Coder. For example, the command line: **C:\B-Coder\B-Coder.EXE C:\B-Coder\MyConfig.BCF** would cause B-Coder to automatically read in and use the configuration file "MyConfig.BCF" at startup. B-Coder's SETUP program also creates an association between the file name extension .BCF and the executable file B-Coder.exe. What this means is that you can also execute B-Coder configuration files directly. For example if you double click on a valid B-Coder configuration file name (with the extension ".BCF") in the Windows Explorer, Windows will automatically launch B-Coder and load in the selected configuration.

The **New** command in the File Menu causes B-Coder to reset itself to its default configuration.

The **Open Configuration** command allows you to open a previously saved B-Coder configuration file. When you select this option, a standard Windows file selection dialog box will appear so you can easily select a previously saved B-Coder configuration file. When you open a configuration file, the filename will appear in the title bar of the B-Coder window.

The **Save Configuration** command allows you to quickly save any changes that you make to an open configuration and the **Save Configuration As...** command allows you to save your current B-Coder configuration with a filename that you specify. When you choose **Save Configuration As...**, a standard Windows "file save as" dialog box will appear so that you can enter the filename and directory path where you would like your configuration to be saved.

The next three commands in the File menu: **Read Message From File...**, **Open Message File...** and **Close Message File** allow you to open and close bar code message files. Instead of entering bar code messages by typing data into the bar code message text field in B-Coder's window, B-Coder can read in data from disk files and use that data as the source of your bar code messages.

The menu command **Read Message From File...** allows you to read in the entire contents of a disk file into the Bar Code Message text field. This option is typically used with the two dimensional symbologies including PDF417, Aztec Code and Data Matrix. Because the two dimensional symbologies can encode up to 3700 characters in a single bar code, a typical two dimensional bar code application might be to read in the contents of a disk file instead of manually typing in all the data for a symbol.

Note: B-Coder has a built in text editor available when any of the two dimensional symbologies are selected in the **Symbology** menu. This text editor is capable of reading data from disk files however, it is designed

for editing text data only and cannot be used for editing files that contain characters that cannot be typed in on your keyboard. (i.e. binary files or data containing ASCII control codes.) If you need to read in disk files containing binary data, you should use the *Read Message From File...* option in the File menu from B-Coder's main menu and not the *File - Open* menu command in the B-Coder text editor. See Also: Symbology Descriptions And Rules for, and Data Matrix (pg. 36, 43, 46).

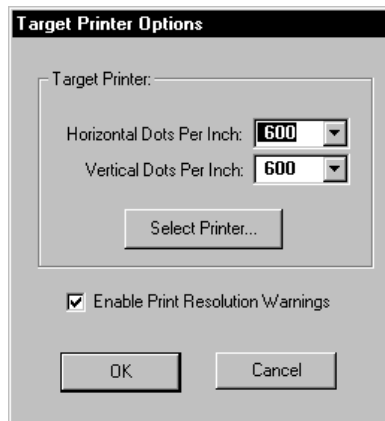
The command **Open Message File...** allows you to open a file that contains multiple bar code messages. When you open a message file, B-Coder reads in each line of the file as a separate bar code message. For example, you could create a bar code message file very easily using the "Notepad" text editor that comes with Windows. Each line of data that you enter in Notepad would be treated by B-Coder as a separate bar code message.

When you open a bar code message file, the first line of data in the file will be displayed in the bar code message text field in B-Coder's window and two buttons labeled **Next** and **Prev** will appear in the toolbar below B-Coder's main menu. To scroll forward or backward through the messages in the file simply click the Next or Prev buttons or press function key **F5** to scroll forward or function key **F4** to scroll backward. The **Edit** menu commands **Next Message** and **Previous Message** can also be used to scroll forward and backward through a message file.

The **Close Message File** menu command is used to close an open message file when you are finished or when it no longer needs to be open.

The **Target Printer** command allows you to specify the dots per inch resolution of the printer that you will use to print your bar codes on. The reason that B-Coder allows you to select the target printer is because B-Coder can be configured to warn you if you try to create a bar code that will not be readable when printed on a particular printer. For example if the target printer has a resolution of 100 dots per inch (i.e., 10 mils per dot) and you try to create a bar code with a narrow bar width less than the width of three printer dots, B-Coder will warn you that the bar code may be unreadable when printed on the selected target printer. B-Coder also allows you to specify all dimensions for your bar codes in units of printer dots instead of inches and mils or metric units. Specifying all bar code dimensions in printer dots guarantees that you will end up with perfectly readable bar codes no matter what type of printer you will be using. See Also: Bar Code Dimensions (pg. 22) and Preferences Menu Commands (pg. 16)

When you select **Target Printer** from the File menu, the following dialog box will appear:



The values that appear by default for the horizontal and vertical dots per inch will be those of the currently selected default printer. If the default printer is not the printer that you intend to use and you know the horizontal and vertical dots per inch values for the printer that you will be using, then you can enter those values manually by typing them in. If you do not know the correct values for a particular printer but a Windows printer driver has been installed on your system for the desired printer, then you can use the **Select Printer...** option to select the specific printer from a standard Windows printer selection dialog box. When you select a printer using the Printer Setup option, B-Coder will automatically calculate the horizontal and vertical dots per inch values for the chosen printer.

The check box labeled **Enable Print Resolution Warnings** is used to enable or disable printer resolution warning messages. This option is also available in the **Preferences** menu.

The File menu command **Printer Setup...** will open the Printer Setup dialog box for the currently selected default printer. The dialog box that will appear will have options that are specific to the currently selected printer.

The File menu command **Print...** will print any bar code graphic that is currently displayed in the B-Coder window. The Print... option is provided mainly so that you can test the printed output directly from within B-Coder. When you select this option a dialog box will appear prompting you for a Left and Top margin value in inches. The margin values specify exactly where on a page the bar code should be printed. When you print a bar code from within B-Coder, the bar code is printed as a Windows MetaFile and therefore should produce the best output possible for whatever printer you are using.

The File menu command **Save Bar Code As...** opens a standard Windows "File Save As..." dialog box so that you can save your bar code images to a disk file. B-Coder allows you to save bar codes in most standard graphic file formats including the vector formats: EPS and WMF and in the raster image formats BMP, PNG, GIF, JPG, .TIF, TGA and .PCX. The raster graphic formats allow you to select various options that affect how the image is actually stored on disk including the Dot Resolution and the Bits Per Pixel (number of allowable colors). The File menu command **Raster Image Properties....** opens a dialog box where you can set the specific properties for all raster style images. For more information on the different options available for all raster style images see: The Raster Image Properties Dialog Box (pg. 11))

To save a bar code in a particular graphics format, specify the file name along with the normal three letter file name extension for the particular format that you want to use. For example, to save a bar code as an EPS file, specify a file name with the extension **.EPS**.

Important: B-Coder initially creates all bar code graphics as Windows **MetaFile** graphic images. MetaFiles consist of a sequence of detailed drawing instructions that are used by Windows programs to reproduce a graphic image exactly as it was originally created. Because of this, MetaFiles are both smaller and much more accurate than any other type of graphic representation. When you save a bar code to a disk file in a different graphic file format, B-Coder will convert the graphic from the original MetaFile format to the specific format that you choose. Because other graphic formats generally have lower resolutions than a MetaFile, images can get distorted when they are converted and thus your bar codes may not always be readable. Also, the file size of most other image formats are usually much larger than an equivalent MetaFile. **It is highly recommended that you save your bar codes as MetaFiles whenever possible.**

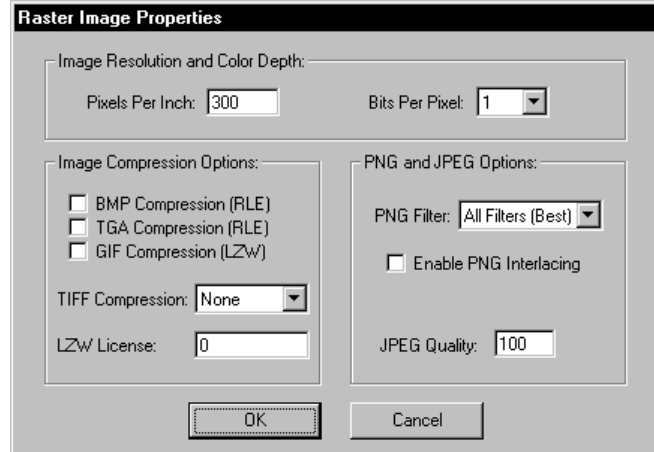
The command **Exit** in the File menu will close B-Coder's window and remove it from memory.

The Raster Image Properties Dialog Box

B-Coder has the ability to save bar code images to a disk file in several different graphic file formats including the raster image formats .BMP, .JPG, .PNG, .GIF, .TIF, .PCX and .TGA. Raster style images consist of an array of discrete pixel values where each pixel (picture element) has a value that represents the color of the pixel. The width of a raster image is defined by the number of pixels across a row. The height of the image is defined by the number of rows of pixels. The overall printed dimensions of a raster image is dependent on the dot resolution of the device used to print it. For example if you create a raster graphic that is 300 pixels wide and has 300 rows of pixels and then you print this image on a printer that has a resolution of 300 pixels per inch, you will end up with a printed image one inch square. If you display the same image on a computer screen that has a resolution of 100 pixels per inch, you end up with an image that is three inches square. This means that raster images are "device dependent" where the resolution of the rendering device (i.e. printer or screen) must be taken into consideration when you create the image. If you need to preserve the size of an image from one device to another, (i.e. screen to printer) you must "stretch" or "shrink" a raster image to fit the desired size. The process of stretching and shrinking involves either adding or removing pixels to or from the original image. As you can imagine resizing an image to a desired size when moving from one output device to another generally causes degradation of the original image quality. Therefore, when creating precise graphics (like bar codes) it is extremely important that the image is created with a fairly large number of pixels per inch (i.e. a high resolution). The best value to use is the pixel resolution (sometimes called the Dots Per Inch or DPI) of the printer that you will be using to print your bar codes.

B-Coder allows full control over the number of Pixels Per Inch (DPI) in all raster images as well as the number of bits per pixel (the number of possible colors). B-Coder also supports several compression options for different raster image formats. Before you save an image to a disk file you can review or change the number of dots per inch, the number of bits per pixel and the compression options so that you end up with the best quality raster images for your particular application.

The File menu in B-Coder contains an option with the caption "Raster Image Properties..." that will display the following dialog box when selected.



The Pixels Per Inch option allows you to specify the number of pixels per inch in the image (the Dot Resolution). The default value is 300 which should provide a fairly high quality image. B-Coder will allow you to set the Pixels Per Inch to any value between 20 and 1440. The higher the number, the better resolution the image however the size of the image file will grow significantly with larger values. If you specify a value that is greater than the number of pixels per inch that your printer is capable of rendering, then the extra resolution will not increase the quality of the printed image. Because of this, you should not specify values that are higher than the dot resolution of your printer.

The Bits Per Pixel option is used to specify the color depth (number of possible colors) in the image. Most raster image formats can have either 1, 8 or 24 bits per pixel. One bit per pixel allows for only two colors in the image, 8 bits per pixel allows for up to 256 colors and 24 bits per pixel allows for up to 16 million colors. The one bit per pixel option will produce the smallest file size for your images. Increasing the value will likewise increase the size of your images proportionally.

Because most bar codes consist of only black bars and white spaces, the one bit per pixel option is the best choice for most situations.

Note: GIF and PCX files only support one or eight bits per pixel, JPG files only support eight or 24 bits per pixel and TGA files only support 24 bits per pixel. If you select a bits per pixel value that is not supported for a particular graphic format and then save a bar code to a disk file using that format, B-Coder will automatically adjust the Bits Per Pixel value to the nearest supported value.

The Image Compression Options allow you to enable different types of compression schemes for different graphic file formats. Enabling compression can dramatically reduce the size of your image files however some application programs do not support the ability to import compressed images. BMP and TGA files support a compression scheme called Run Length Encoding (RLE). TIF files support several different compression schemes including LZW, PackBits, Group3 and Group4. The choice of which to use will depend on the graphics program that you intend to use your bar code images in. GIF files are normally compressed using the LZW compression scheme however uncompressed GIFs can also be created. Unisys Corporation holds a patent on the LZW compression algorithm and requires you to pay a license fee of between \$5000 and \$7500 in order to create compressed GIF files. B-Coder supports the ability to create compressed GIF files however you need to obtain a LZW license from Unisys in order to enable this functionality.

If you purchase a LZW license from Unisys, you can send a copy of your license to TAL Technologies, Inc. and we will provide you with a key number that you can enter in the LZW License text field in the Raster Image Properties dialog box to unlock the LZW compression capabilities in B-Coder. For information concerning licensing the LZW compression and/or decompression capability, please contact: Unisys Corporation, Welch Licensing Department - C1SW19, Township Line & Union Meeting Roads, P O Box 500, Blue Bell PA 19424. Web Site: <http://www.unisys.com/unisys/lzw/>

Note: Microsoft has entered into an agreement with Unisys so that they can create compressed GIF files in the Paint program that they supply with Windows. One technique that you can use to create compressed GIF files legally with B-Coder is to save your bar code images to a disk file as uncompressed GIF files in B-Coder and then import the image file into the Paint program that comes with Windows and finally save the file as a GIF from the Paint program.

The PNG file format is a newer graphic format that was designed to produce highly compressed image files. The PNG format supports several different filter options that determine how an image is compressed. The PNG Filter option in B-Coder allows you to choose a specific filter option for PNG files. The default for this option is "All Filters" which will use all of the available filters to produce the smallest image. PNG files also support a feature called Interlacing that cause the data in the image to be saved in an interlaced manner. The idea behind interlacing is that the image data is stored in the file in a manner so that a web browser can fill in the image as it is being downloaded from a web server so that the image can be displayed almost immediately with more detail being rendered as the data is being downloaded. If you will be creating bar codes that will be downloaded and displayed in a web browser, you might want to enable the PNG Interlacing option.

JPEG images use a compression scheme that is adjustable however the more compression that is used, the more detail is lost from the original image. The term used to describe this is called "JPEG Quality" and it is referred to in terms of a percentage from 1 to 100 with 100 being the best quality. For almost all bar code applications it is extremely important that you maintain the highest quality images in order to be readable by a bar code scanner. Unless you have a specific reason to change the JPEG Quality, you should leave it set to its default value of 100%.

Edit Menu Commands

Edit	
Build/Copy	F2
Paste	Ctrl+V
Incremental Numbering...	
Next Message	F5
Previous Message	F4

The **Edit** menu contains the choices: **Build/Copy**, **Paste**, **Incremental Numbering...**, **Next Message** and **Previous Message**.

The **Build/Copy** command is used to generate a bar code from the text entered in the bar code text edit box in B-Coder's window. When you select Build/Copy, your bar code will be generated, displayed, and finally the graphic image of the bar code is copied into the Windows clipboard in the Windows Metafile graphic format so that you can paste it into other Windows programs.

Note: Some newer 32 bit Windows programs including Word, Pagemaker and Excel will automatically convert Windows Metafiles to a newer "Enhanced Metafile" format when they are pasted through the Windows clipboard. Unfortunately, the conversion process will generally degrade quality of your bar codes. These programs still support the Windows Metafile format however in order to bypass the conversion to the Enhanced Metafile format, you will need to paste your bar codes into these programs by selecting the "Paste Special" option in these programs and then select either "Picture" or "Windows Metafile" from the Paste Special dialog box. For more information see: Pasting Bar Codes Into 32 bit Programs (pg. 28).

Hint: Double clicking your *left* mouse button with the mouse cursor on any blank space in B-Coder's window will perform the same function as selecting Build/Copy from the Edit menu.

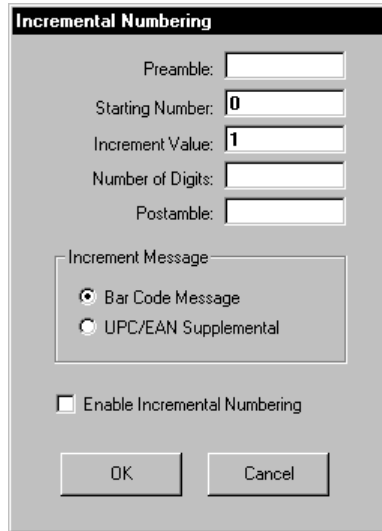
The **Paste** command pastes text from the Windows clipboard into the Bar Code Message text box or the Comment Line text box. Thus you can paste text strings from other programs into B-Coder. The Paste option will be active (not grayed) whenever there is text data in the clipboard.

NOTE: When you create a bar code by selecting Build/Copy, the clipboard is first cleared and the graphic image of your bar code is placed there. After a Build/Copy operation, the clipboard will no longer contain text data and the Paste feature will be deactivated until another text string is "Copied" from another Windows program.

The **Incremental Numbering...** command in the Edit menu opens a dialog box where you can specify all parameters necessary to have B-Coder automatically create a sequence of bar code messages containing a number to be incremented by a fixed amount.

For example, suppose you need to create a set of bar code labels whose messages are a series of numbers that are incremented by a fixed amount. The Incremental Numbering option allows this type of task to be performed simply and quickly.

When you select **Incremental Numbering...**, the following dialog box will appear:



The dialog box titled "Incremental Numbering" contains the following fields and options:

- Preamble: []
- Starting Number: [0]
- Increment Value: [1]
- Number of Digits: []
- Postamble: []
- Increment Message section:
 - Bar Code Message
 - UPC/EAN Supplemental
- Enable Incremental Numbering
- Buttons: OK, Cancel

B-Coder allows you to specify five parameters to define incrementally numbered bar code messages: a **Preamble**, a **Starting Number**, an **Increment Value**, the **Number of Digits**, and a **Postamble**.

The **Starting Number** is a numeric value that B-Coder will use as the first number in the sequence. The **Increment Value** is the value that is to be added to the Starting Number for each new number in the sequence.

The **Number of Digits** parameter is used to optionally specify the number of digit positions for the incremented value. For example, if you specify a starting number of 100, an increment value of one and set the Number of Digits to five, then the sequence that is generated would be: 00100, 00101, 00102, etc... Because the Number of Digits was set to five, B-Coder automatically left pads the number with zeros forcing the resulting string to be five digits wide. If the Number of Digits parameter is left blank or set to zero then no padding or zero filling is performed.

The **Preamble** is a text string that immediately *precedes* the number that will be incremented. The **Postamble** (similar to the Preamble) is a text string that immediately *follows* the number to be incremented. For example, if the Preamble is defined as: **Fried** and the Postamble is defined as: **Chicken** in the example above, the resulting sequence of numbers would be: Fried00100Chicken, Fried00101Chicken, etc...

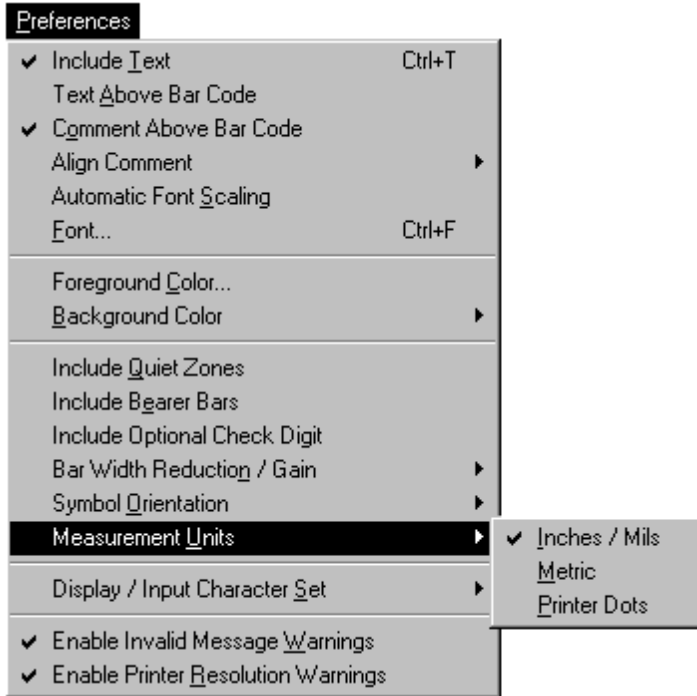
Note: If you press the Enter key while editing a preamble or postamble, a carriage return linefeed will be inserted in the text (appearing as a musical note and a rectangle with a circle in it). Unless you want a carriage return linefeed in your preamble or postamble, you should not press the Enter key when finished editing either of these fields. Also, if the currently selected symbology supports characters that cannot be typed in on your keyboard, an ASCII chart will appear in the incremental numbering dialog box so that you can enter all characters in the symbology's character set by selecting them from the ASCII chart.

B-Coder can be configured to increment either the bar code message text or the supplemental text for UPC or EAN type bar codes. For information about UPC and EAN supplemental messages, see the section Bar Code Symbology Descriptions and Rules (pg. 30).

You can choose which message to apply the incremental numbering to by selecting either **Bar Code Message** or **UPC/EAN Supplemental** from inside the frame marked **Increment Message**.

The check box marked **Enable Incremental Numbering** is used to activate or deactivate incremental numbering. When incremental numbering is active, two buttons labeled **Next** and **Prev** will appear in the toolbar below B-Coder's main menu. To scroll forward or backward through the incremental messages simply click the Next or Prev buttons or press function key **F5** to increment forward or function key **F4** to decrement backward. The **Edit** menu commands **Next Message** and **Previous Message** can also be used to scroll forward and backward through the incremental values.

Preferences Menu Commands



Most of the commands in the Preferences menu except the **Font**, **Foreground Color**, **Background Color** and any advanced option commands are toggle options. Selecting any of the toggle options will switch the option either ON or OFF where "ON" is indicated by a check mark appearing to the left of the menu item.

The **Include Text** option causes B-Coder to include the human readable version of your bar code message neatly centered along the bottom or top edge of the bar code symbol.

Note: This option does not apply to PDF417, Aztec, Data Matrix, POSTNET, or BPO 4 State bar codes.

The **Text Above Bar Code** option causes B-Coder to print the human readable version of your bar code message along the top edge of the bar code symbol. If this option is not selected, B-Coder will print the message along the bottom edge of the bar code symbol.

The **Comment Above Bar Code** option causes the comment text to be printed above your bar code. If this option is not selected, the comment will be printed below the bar code symbol.

The **Align Comment** option allows you to choose to have the comment text aligned with either the Left, Center or Right edge of the bar code symbol.

The **Automatic Font Scaling** option causes B-Coder to automatically scale all text to a font size that is approximately one fifth the height of the generated bar code symbol.

The **Font...** command displays a standard font selection dialog box where you can choose the font for all text printed with a bar code symbol. The font selection dialog box allows you to select a font name, size, color and strikethrough or underline effects for a chosen font.

NOTE: Most bar code symbology specifications recommend the font **OCR-B** (Optical Character Recognition revision B). The choice of font is not really that important unless your bar codes are being used in an OCR application however it is a good idea to choose fonts that are close to the recommended specification. The font **MS Sans Serif** that comes with Windows is very close to OCR-B and the True Type Font **Arial** is also very similar.

Different fonts behave differently thus some fonts may appear different on screen than when printed. True Type fonts are the most WYSIWYG under Windows and they also align better when rotated 90 or 270 degrees.

Note: When the Automatic Font Scaling option is selected, it will override any font size chosen from the font selection dialog box

The **Foreground Color** and **Background Color** options allow you to specify the colors used for the bars and spaces in your bar code symbols. The Background Color option can be set to either **Transparent** or to a specific **Custom Color**. When you select either Foreground Color or a Custom Background Color, a standard Windows color selection dialog box will appear where you can select a desired color from a color palette. The color palette has an option allowing you to define custom colors by specifying exact Red, Green and Blue (RGB) color values or by specifying colors in terms of Hue, Saturation and Luminescence values.

Note: It is entirely possible to choose color combinations that render a bar code symbol unreadable. The reason a bar code reader is able to read a bar code at all is because of a high contrast between the shades of the bars and spaces. Although two colors may appear to the human eye to have a high of contrast between them, a bar code reader may not be able to determine any difference at all between the two colors. Solid black bars on a solid white background always produces the best results. If you must use colors other than black on white, a good rule of thumb is to select solid foreground colors with a luminescence value no greater than 60 and select solid background colors with a luminescence value no less than 180. **Always test colored bar codes** before committing to a color scheme.

See Also: Bar Code Basics -How A Bar Code Reader Works (pg. 26)

The **Include Quiet Zones** command causes B-Coder to include a small region of blank space (10 times the width of the narrow bar elements) at either end of a bar code image. This space, called *Quiet Zones*, helps to insure that a bar code reader will be able to correctly determine the true beginning and end of a bar code symbol. Most bar code symbology specifications require quiet zones therefore it is highly recommended that you enable this option.

The **Include Bearer Bars** command causes B-Coder to draw lines around a bar code symbol. The purpose of bearer bars is to equalize the pressure exerted by a printing plate over the entire surface of the symbol. Bearer bars also enhance the reading reliability of a bar code by reducing the probability of misreads or short scans which may occur when a skewed scanning beam enters or exits the symbol through the top or bottom edge of the bar code. When the scanner path leaves the symbol through the top or bottom, it crosses the bearer bar, thereby resulting in an invalid start/stop code. Bearer Bars are typically used with Interleaved 2 of 5 and EAN/UCC 128 type bar codes.

The **Include Optional Check Character** command causes B-Coder to automatically calculate and append an additional check character to all Code 39, Codabar, Interleaved 2 of 5 or MSI Plessey bar code symbols. This feature should be enabled if the bar code reader that you are using is configured to expect a check character when reading Code 39, Codabar, Interleaved 2 of 5 or MSI Plessey bar codes. This feature only applies to Code 39, Codabar, Interleaved 2 of 5 or MSI Plessey type bar codes. Code 39 with a check character is also sometimes referred to as **HIBC** Code. (Health Industry Bar Code) The check character adds an extra measure of security to a bar code symbol for more "mission critical" applications. Codabar, Interleaved 2 of 5 or MSI Plessey bar code applications rarely require a check character.

The **Bar Width Reduction / Gain** option allows you to specify a Reduction or Gain factor ranging from 0% to 75%. Selecting a non-zero Bar Width Reduction or Gain causes B-Coder to reduce or enlarge the width of all solid bars in a bar code. Bar Width Reduction is often necessary to compensate for ink spread when generating bar codes that will be used in wet ink printing processes. The percentage that you specify is based on the narrow bar width that you choose for your bar codes. For example if you specify a bar width reduction of 25% and your narrow bar width is set at 10 mils, B-Coder will reduce the width of all solid bars in your bar codes by 2.5 mils (25% of 10 mils = 2.5 mils). Bar width gain is typically used when printing on glass or other surfaces that cause ink to bead up or shrink as it dries.

The **PDF417 Options...** command will display a dialog box where you can specify several parameters that control the way that B-Coder generates PDF417 bar code symbols. For details about all the features and capabilities of the PDF417 symbology as well as the PDF417 options available in B-Coder, see the section: Symbology Descriptions and Rules for PDF417 (pg. 36).

Note: The Aztec Code and Data Matrix symbologies also support special user selectable options. When either of these symbologies is selected in the Symbologies menu, this menu option will change to either **Aztec Code Advanced Options** or **Data Matrix Advanced Options** so that you can choose the advanced options for these symbologies.

See Also: Symbology Descriptions and Rules for Aztec Code and Data Matrix (pg. 43, 46).

The **Symbol Orientation** option allows you to rotate a bar code symbol in increments of 90 degrees from horizontal. Three choices of orientation are available: Horizontal (picket fence orientation), Vertical 90 degrees clockwise and Vertical 270 degrees clockwise (ladder orientation).

The **Measurement Units** option allows you to select either **Inches/Mils**, **Metric units** or **Printer Dots** for all bar code dimensions. Bar code dimensions are normally expressed in Inches and Mils however B-Coder lets you have your choice of measurement units. When creating bar codes with very small dimensions, the best choice of units is Printer Dots. When bar code dimensions are expressed in Printer Dots, the resulting printed symbols will have the least possible chance of being distorted due to limitations imposed by the dot size resolution of the printer being used. A printer cannot print lines narrower than the width of a single printer dot therefore if you specify dimensions that are not integer multiples of the width of a printer dot, you could end up with distorted or unreadable bar codes.

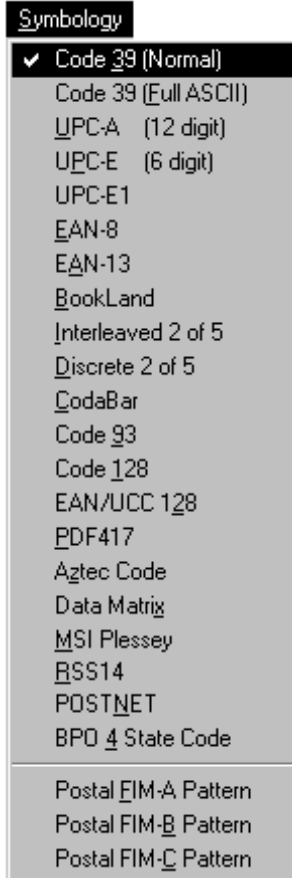
The **Display/Input Character Set** option allows you to change the character set that B-Coder uses for the Bar Code Message and is provided to support international characters when using the PDF417, Aztec Code or Data Matrix bar code symbologies. The two choices for this option are ASCII or ANSI. Because PDF417, Aztec and Data Matrix support the entire 256 byte ASCII (or ANSI) character set, accented or other special international characters can be encoded in these types of bar codes. The ASCII and ANSI character sets are very similar however the ANSI character set is better suited for international characters. The choice of which character set to use will depend primarily on the application that you will be reading your bar codes into after they are scanned by a bar code reader. If the application is expecting Extended ASCII character data then you should select the ASCII character set. Likewise, if the program is expecting ANSI character data then you should select ANSI as the character set in B-Coder.

When you choose between the ASCII and ANSI character sets in B-Coder, all that B-Coder actually does is to change the font used to display the text that you enter into the bar code message edit field in the B-Coder window. If you choose ASCII as the character set then B-Coder uses the Terminal font which uses the Extended ASCII character set. If you choose the ANSI character set then B-Coder uses the Arial font which likewise uses the ANSI character set.

The option **PDF417 Symbol Character Set** allows you to change the character set that B-Coder uses when generating PDF417 bar code symbols. The three choices provided for this option are "Display Character Set", "ASCII" or "ANSI". The default for this option is "Display Character Set" which means that B-Coder will use the character set that is selected as the "Display/Input Character Set" when generating PDF417 bar code symbols (see above). The other two options "ANSI" and "ASCII" are used when you want to convert a bar code message from ANSI to ASCII or from ASCII to ANSI before generating a PDF417 bar code symbol. For example, suppose you have selected "ANSI" as the "Display/Input Character Set" but you want to generate PDF417 bar codes for a system that uses the ASCII character set. If you select "ANSI" as the "Display/Input Character Set" and you select "ASCII" as the "PDF417 Symbol Character Set", B-Coder will perform an ANSI to ASCII character conversion on your bar code message data before generating any PDF417 bar code symbols.

The **Enable Invalid Message Warnings** and **Enable Printer Resolution Warnings** options cause B-Coder to warn you with a message box whenever you try to generate a bar code that does not conform to the rules for the currently selected symbology or when a bar code will not print correctly on the selected target printer. When these options are disabled, B-Coder will not prompt you with a warning but it will beep your PC's speaker one short beep. These options are typically disabled when B-Coder is being controlled by another application through the use of DDE commands. See Also: DDE Commands Supported By B-Coder (pg. 51)

Symbology Menu Commands



The SYMBOLOGY menu allows you to select a specific symbology for your bar codes. The currently selected symbology will have a check mark to the left of its name in the menu.

For details about each of the symbologies supported by B-Coder see the section: Bar Code Symbology Descriptions and Rules (pg. 30)

When you select a particular symbology, the bar code dimension data entry fields in B-Coder's window may change to reflect the options available for the specific symbology.

See Also: Bar Code Dimensions (pg. 22) and Bar Code Symbology Descriptions and Rules (pg. 30)

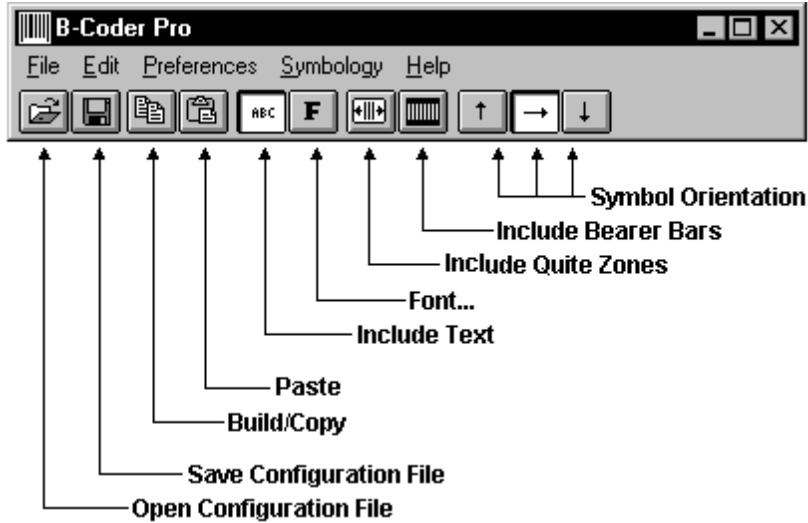
Help Menu Commands

The HELP menu contains the choices: *Index*, *TALtech On the Web* and *About*.

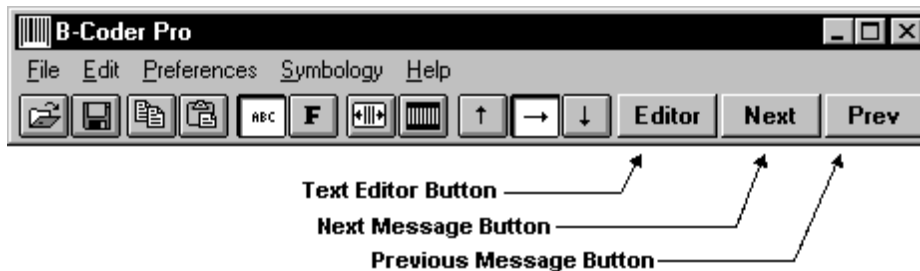
The *Index* command accesses the Windows Help system and displays an index for all help topics available for B-Coder and the *About* command displays a product version number and a copyright notice. The *TALtech On the Web* command will open your Internet web browser to the TALtech home page where you can obtain the most up to date technical support information for B-Coder.

The B-Coder Toolbar

The toolbar below B-Coder's main menu provides quick access to some of the more useful commands in B-Coder's main menu. Below is a summary of the main menu commands that are associated with each toolbar icon. Clicking your mouse on any of the tool icons in the toolbar has the same effect as selecting the associated command from the main menu.



If a Bar Code Message File is open or if Incremental Numbering is enabled, two buttons marked **Next** and **Prev** will also appear that allow you to scroll forward or backward through a series of bar code messages. When PDF417, Data Matrix or Aztec is the current symbology, a button marked **Editor** will appear that allows access to a text editor for editing long bar code messages.



Bar Code Dimensions

Because of differences in the design of each bar code symbology, there are differences in the way that the dimensions for each symbology are expressed. When you select different symbologies from B-Coder's symbology menu, you will notice that some of the data entry fields for entering the dimensions of your bar code will change. For example, when you select Code 39, B-Coder allows you to specify a Bar Code Height, a Narrow Bar Width, and a Narrow To Wide Ratio. When you choose UPC or EAN type bar codes, B-Coder allows you to specify a Bar Code Height, a Narrow Bar Width, and a Magnification Factor.

The two main dimensions used to define the size of most linear bar codes are the **Narrow Bar Width** and the overall **Bar Height**. The Height is generally less important than the Narrow Bar Width and you can scale the height to any size that you like. For the sake of readability the height should not be less than a quarter of an inch or 15% of the total width of the symbol, whichever is greater. UPC and EAN bar codes have more specific size requirements if they are to be used for product identification. For details refer to UPC and EAN Magnification Factors on the following page.

The Narrow Bar Width effectively determines the total width of a bar code symbol. All other bar and space width dimensions are based on this width (referred to as the "nominal X dimension"). The best choice for this dimension depends partly on the resolution of your bar code reading equipment and also on the resolution of the printer being used to produce the bar code.

As a general rule the Narrow Bar Width should fall in a range between 10 to 30 mils and should never be less than 7.5 mils. For UPC and EAN bar codes, the smallest allowable Narrow bar width is 10.4 mils. One simple way to determine a good Narrow Bar Width is to actually print out a typical bar code using several different values and try reading each one with your scanning equipment. You should choose the value that produces bar codes with the highest "first pass" read rate.



Narrow To Wide Element Ratios

Some symbologies consist of bars and spaces with only two element widths, *Narrow* and *Wide* Elements where the width of the wide elements is a multiple of the width of the narrow elements. The specifications for these symbologies allow you to choose a *Narrow to Wide Element Ratio*.

The symbologies that support this feature are Code 39, Interleaved 2 of 5, Discrete 2 of 5, and Codabar. If you select one of these symbologies, a list box will appear where you can select a narrow to wide element ratio between 2.0 and 3.0 in increments of .1 (i.e. 2.0, 2.1, 2.2, 2.3....3.0). The rules for these symbologies specify that when the narrow element width is less than 20 mils, the narrow to wide element ratio must be 2.2 or greater. B-Coder defaults to a narrow to wide ratio of 2.5 and this value should be acceptable for most applications.

Note: Higher quality readers may be able to successfully read bar codes with a narrow to wide ratio less than 2.2 no matter what the narrow element width is. Lower quality readers often need a ratio of at least 2.5 in order to read correctly. Because of the great variability between readers, you should always experiment with different ratio values and select the value that produces bar codes with the best successful read rate.

UPC and EAN Magnification Factors

UPC and EAN bar codes are defined with a standard, fixed, narrow element width of exactly 13 mils; however, the specifications for these symbologies allows for a *magnification factor* of between 80% and 200%. This translates to an allowable range of narrow element widths of between 10.4 and 26 mils.

When you select either of these symbologies a text box will appear where you can specify the magnification factor as a percentage of the standard element width (80% to 200%).

See Also:

Bar Code Symbology Descriptions & Rules for UPC and EAN (pg. 31-32)
How To Produce Readable Bar Codes (pg. 25)

PDF417 Bar Code Dimensions

A PDF417 bar code symbol consists of multiple rows of bar code data encoded in units called *code words*. Each symbol can contain from 3 to 90 rows and each row consists of a Start/Stop pattern and from 3 to 32 code words (1 to 30 code words for data and 2 for Right and Left Row Indicators). The smallest element in a PDF417 symbol is called a **module**. Each code word consists a unique pattern of 4 bars and 4 spaces each with a width of up to 6 modules within a total width of 17 modules per code word. (This is where the 417 comes from in the name PDF**417** - 4 bars within 17 modules.)

Because the number of code words in a row and the total number of rows in a symbol are variable quantities, the three primary dimensions used to define the size of a PDF417 bar code are the **Module Width**, the **Module Height**, and the overall **Aspect Ratio**. The Module Height and Width values define the height and width of the smallest element in the PDF417 symbol and the Aspect Ratio specifies the overall shape of the symbol.

The best choice for the **Module Width** dimension depends partly on the resolution of your bar code reading equipment and also on the resolution of the printer being used to produce the bar code. The specification for PDF417 recommends that the Module Width should fall in a range between 10 and 30 mils. The smallest allowable module width defined in the symbology specification is 6.56 mils. (This translates to 2 printer dots when printing to a 300 DPI laser printer.) The best way to determine the ideal Module Width for your application is to actually print out a sample bar code using several different values and try reading each one with your scanning equipment. Again, you should choose the value that produces bar codes with the best read rate.

The recommended value for the **Module Height** is approximately three times the value for the Module Width however the symbol specifications allow for module heights as small as 10 mils (3 printer dots on a 300 DPI laser printer).

The **Aspect Ratio** determines the overall shape of the PDF417 symbol and is defined as the overall height to width ratio. Higher values for the Aspect Ratio (greater than 1) produce tall, thin PDF417 bar codes and small values (greater than zero and less than 1) produce short, wide bar codes. A value of 1 should produce approximately square bar codes. Because each row of data in a PDF417 symbol contains 2 additional code words of overhead (right and left row indicators) as well as start and stop patterns, tall and thin bar codes with lots of rows (high aspect ratios) will be larger in total area and contain more overhead than short wide bar codes (low aspect ratios). On the other hand, short wide bar codes (low aspect ratios) may be difficult to read due to scanner width limitations.

NOTE: Because of the basic structure of a PDF417 symbol, you will probably not be able to produce a bar code with the exact Aspect Ratio that you specify. B-Coder will automatically generate a bar code with the closest match to the Aspect Ratio that you specify within the limits of the symbol specification.

For any given PDF417 bar code symbol there are only a small number of aspect ratios that are physically possible. When you generate a PDF417 bar code symbol, B-Coder will automatically calculate all possible aspect ratios for the given bar code message and then provide them in a drop down list box below the Aspect Ratio data entry field in the B-Coder Window. After you generate a PDF417 symbol, you can experiment with all the different possible Aspect Ratios for a specific message by selecting different values from the list created by B-Coder.

B-Coder also provides more detailed control over the size and shape of a PDF417 symbol by allowing you to specify the maximum number of rows and a maximum number of data code word columns in a symbol. You can also cause B-Coder to produce a version of PDF417 called **Truncated PDF417** that removes the Right Row Indicator code words and the stop pattern on the right hand side of the symbol thus reducing the total size of a PDF417 symbol. These options are available by selecting **PDF417 Options...** from the Preferences menu..

See Also: Symbology Descriptions and Rules for (pg. 36)

Aztec Code and Data Matrix Bar Code Dimensions

Aztec Code and Data Matrix bar code symbols are built up into a rectangular grid or "Matrix" of square dots or "Modules". The module size is the only dimension that can be explicitly chosen by the user. The overall size of the bar code symbol is dependent on the module size, the amount of data encoded in the symbol and the amount of error checking and correction that you choose. Because of the way most 2-D bar code readers work, the scanner manufacturers recommend that you use a module size of between 15 and 30 mils with 20 mils being the preferred module size.

How To Produce Readable Bar Codes

Although there are many different types of bar codes, they all share the same requirements in order to be readable by most commercially available scanning devices. Because a beam of light is used to read a bar code, it should be clean and free of defects or smudges and there should be a high contrast between the color of the bars and the color of the spaces. Black bars on a white background yield the best results. If you intend to use colored bar codes or colored paper, you should always test the readability of your bar codes before committing to a color scheme.

Another important consideration is that there should always be a small amount of space or *Quiet Zones* preceding and following the bar code so that the reading device is able to properly determine the true start and end of the bar code symbol. A good rule of thumb is to reserve at least a quarter of an inch or 10 times the width of a single narrow bar (whichever is greater) for blank space at either end of a bar code. B-Coder will automatically include Quiet Zones of ten times the narrow bar width if the "Include Quiet Zones" option is enabled in the Preferences menu.

When printing bar codes, laser, ink jet and thermal transfer printers produce the best results. Dot matrix printers produce the poorest quality (but not necessarily unacceptable) bar codes. Dot matrix printers are especially poor when the bar code dimensions are set to a small size. You should try to avoid very small or very large bar codes, both narrow bar widths and overall bar code dimensions. As a precaution you should always test your printed output with whatever bar code reading equipment you eventually intend to use.

B-Coder can be configured to display a warning message if the specified Narrow Bar Width will be smaller than three printer dots for a particular printer. You can select a specific target printer or printer resolution and also enable Printer Resolution Warning Messages using the **Target Printer...** command in the File menu.

See Also:

Bar Code Symbology Descriptions and Rules (pg. 30)
Special Considerations and Incompatibilities (pg. 27)
Bar Code Basics (pg. 26)

Bar Code Basics

Bar code is an automatic identification technology that allows data to be collected rapidly and with extreme accuracy. Because of this, bar code technology is finding its way into a broad range of applications in almost every sector of business. Bar codes provide a simple and easy method of encoding text information that is easily read by inexpensive electronic readers. A bar code consists of a series of parallel, adjacent bars and spaces. Pre-defined bar and space patterns or *symbolologies* are used to code character data into a printed symbol. Bar codes can be thought of as a printed version of the Morse code with narrow bars representing dots, and wide bars representing dashes. A bar code reading device decodes a bar code by scanning a light source across the bar code and measuring the intensity of light reflected back to the device. The pattern of reflected light produces an electronic signal that exactly matches the printed bar code pattern and is easily decoded back to the original data. Due to the design of most bar code symbolologies it does not make any difference if you scan a bar code from right to left or from left to right.

Shown Below is the structure of a typical bar code symbol.

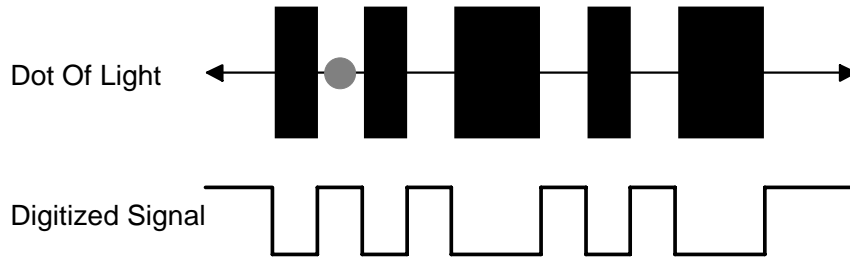


The basic structure of a bar code consists of a leading and trailing quiet zone, a start pattern, one or more data characters, optionally one or two check characters and a stop pattern.

There are a variety of different types of bar code encoding schemes or "symbolologies", each of which were originally developed to fulfill a need in a specific industry. Several of these symbolologies have matured into de-facto standards that are used universally today throughout most industries. The symbolologies supported by B-Coder are those most commonly used across all industries. For more information about specific bar code symbolologies supported by B-Coder see: Bar Code Symbology Descriptions and Rules (pg. 30)

How A Bar Code Reader Works

A bar code reader works by scanning a dot of light across a bar code symbol. As the dot scans across the bar code, light is reflected back to the bar code reader by the light areas and is absorbed by the dark areas. The scanner electronically measures the intensity of the light reflected back to produce a digitized waveform that can be decoded back to the original message similar to the way morse code dots and dashes are decoded.



The factors that make a bar code readable are: an adequate print contrast between the light and dark bars and having all bar and space dimensions within the tolerances for the symbology. It is also helpful to have sharp bar edges, few or no spots or voids, a smooth surface and clear margins or quiet zones at either end of the printed symbol.

Bar code scanners can be purchased with different resolutions to enable them to read bar codes of different sizes. The scanner resolution is measured by the size of the dot of light. The dot of light should be equal to or slightly smaller than the narrowest element width ("X" dimension). If the dot is wider than the width of the narrowest bar or space, then the scanner can get confused due to the dot overlapping two or more bars at a time. If the dot is too small, then any spots or voids in the bars can be misinterpreted as light areas thus making a bar code unreadable.

Special Considerations and Incompatibilities

B-Coder has been thoroughly tested with a wide variety of Windows application programs as well as different printing devices and has been found to produce extremely high quality bar codes. There are a few minor incompatibilities with some programs but fortunately most of these problems have simple solutions. The most common types of problems have been listed below and are outlined in further detail in the following pages.

Pasting Bar Codes Into 32 bit Programs (Word, Excel, Pagemaker 6.x, etc.)
Pasting Bar Codes Into Programs That Work With Bitmaps.
Pasting Bar Codes Into A Spreadsheet.
Printing Bar Codes On A Dot Matrix Printer.

See Also:How To Produce Readable Bar Codes. (pg. 25)

Pasting Bar Codes Into 32 bit Programs (Word, Excel and Pagemaker 6.x, etc.)

When you create a bar code in B-Coder, a Windows MetaFile is placed in the clipboard so that you can paste the bar code into other windows programs. Starting with Windows 95, Microsoft has added a new MetaFile format called the "Enhanced MetaFile" or "EMF" format. The new EMF format is supposed to be compatible with the standard WMF file format however there are some inherent technical problems with the EMF format that make it less than ideal for creating bar codes.

Most 32 bit Windows programs (including Word, Access and Pagemaker) support both the WMF and the EMF file formats however some 32 bit programs (i.e. Word, Excel and all 32 bit versions of Pagemaker) automatically convert any metafile found in the clipboard to an Enhanced MetaFile. When you paste bar codes from B-Coder into these programs by selecting "Paste" from their EDIT menu, the bar code will be converted to an Enhanced MetaFile and subsequently will not print correctly. For these applications you must select "Paste Special" and choose either "Picture" or "MetaFile" (instead of "Enhanced MetaFile") in the "Paste Special" dialog box as the format. Unfortunately Excel does not support pasting metafiles that are not "Enhanced Metafiles" however it will allow you insert metafiles from a disk file as long as the Windows Metafile graphics import filter was installed when you originally installed Excel on your computer. This means that to add bar codes to an Excel worksheet, you must save your bar codes to a disk file using the WMF file format and then use the "Insert Picture From File" menu option in Excel to read the metafile from the disk file into the sheet.

Pasting Bar Codes Into Programs That Work With Bitmaps

When cutting and pasting bar codes from B-Coder into other programs, the best graphics format to use is the MetaFile format. MetaFile images are scaleable images that consist of drawing instructions used internally by Windows to reproduce an image in a manner that uses the highest resolution of the device that the image is being printed on. Because most printers have a much higher resolution than most computer monitors, the printed version of a MetaFile will almost always have a much higher degree of detail than the same image when it is displayed on your computer's monitor. Bitmaps on the other hand do not automatically use the highest resolution possible when printed and therefore the printed output of a bitmap will retain the same (lower) quality as that of your monitor.

When you paste a MetaFile bar code produced by B-Coder into a graphics or drawing program that only has the capability of working with bitmap (or other raster images) such as the **Paint** program provided with Windows, your bar code will be automatically converted from a Windows MetaFile format to a bitmap format. During this conversion, the bar code will be deformed and thus it may be unreadable when it is eventually printed. A possible solution to this problem is to increase the Narrow Bar Width value and avoid the use of any scaling features in the target application (i.e. the Stretch/Skew feature in the Paint program). Acceptable bar codes can be produced with these types of programs but their output should always be tested for readability.

Pasting Bar Codes Into A Spreadsheet

A minor problem with spreadsheets is that they normally display and print a grid pattern around all cells. Cell grid lines can be confused by a bar code reader as part of a bar code. This problem can easily be corrected simply by turning grid lines off before printing the spreadsheet. In Excel this is done by selecting "Options..." from the Tools menu and then selecting the View tab in the options dialog box and finally disabling the "Grid lines" option.

Another small problem is that most spreadsheet programs will automatically draw a border around images both while displayed on screen and when printed on a printer. If there is no leading or trailing "Quiet Zone" at either end of your bar code, the border appears to a bar code reader as part of the bar code and thus it will probably be unreadable. The easiest way to avoid this problem is to enable the **Include Quiet Zones** option in B-Coder's Preferences menu. Another solution is to remove the unwanted border with a formatting command in the spreadsheet. In Microsoft Excel you can get rid of borders around graphics by selecting the bar code image after it has been inserted into the spreadsheet and then choosing "Picture..." from Excel's Format menu. In the "Format Picture" dialog box, select the "Colors and Lines" tab and finally, for the Line Color option choose "No Line".

Printing Bar Codes On A Dot Matrix Printer

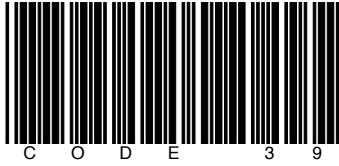
Dot matrix printers offer the lowest resolution of all available printers. Although dot matrix printer manufacturers claim resolutions as high as 360 dots per inch, the real resolution of most is only 60 dots per inch. Higher resolutions are simulated by overlapping consecutive dots. The reason that the true resolution is only 60 dots per inch is because the width of each dot is approximately 1/60th of an inch (16 mils). In order to print readable bar codes on a dot matrix printer, the Narrow Bar Width must never be less than the width of a printer dot because it is impossible for a dot matrix printer to print a line narrower than the width of a single dot. The larger the value that you choose for the Narrow Bar Width (within the allowable range for a specific symbology), the more readable your bar codes will be.

When you set up the Windows printer driver for your dot matrix printer you should select the highest dot resolution that the printer is capable of. If the driver has a Dithering option, you should also select "Line Art" or the finest dithering resolution that is supported. Some dot matrix drivers also allow you to set the intensity or darkness of the printing. If your printer supports this feature, you should set the intensity to the darkest level supported.

Finally, you should always test bar codes printed on a dot matrix printer.
See Also: How To Produce Readable Bar Codes. (pg. 25)

Bar Code Symbology Descriptions and Rules

CODE 39 (Normal and Full ASCII versions)



The Normal CODE 39 is a variable length symbology that can encode the following 44 characters: 1234567890ABCDEFGHIJKLMNPOQRSTUVWXYZ-.*\$/%. Code 39 is the most popular symbology in the non retail world and is used extensively in manufacturing, military, and health care applications. Each Code 39 bar code is framed by a start/stop character represented by an asterisk (*). The Asterisk is reserved for this purpose and may not be used in the body of a message. B-Coder automatically adds the start and stop character to each bar code therefore you should not include them as part of your bar code message. If you select the Normal version of CODE 39 and your bar code text contains lower case characters, B-Coder will convert them to upper case. If your bar code text contains any invalid characters, B-Coder will prompt you with a warning message if the "Enable Invalid Warning Messages" option is selected in the Preferences menu.

Code 39 allows for an optional (modulo 43) check character in cases where data security is important. The health care industry has adopted the use of this check character for health care applications. To enable the Code 39 check character feature in B-Coder, select the option **INCLUDE CODE 39 / I 2 of 5 CHECK CHARACTER** in B-Coder's Preferences menu. When this option is enabled, B-Coder will automatically calculate and append the proper check character to all Code 39 symbols.

Another feature of Code 39 allows for concatenation of two or more bar codes. It is sometimes advantageous to break long messages into multiple, shorter, symbols. If the first data character of a Code 39 symbol is a space, some readers will store the remainder of the symbol in a buffer and not transmit the data. This operation continues for all successive Code 39 symbols with a leading space, with each message appended to the previous one. When a message without a leading space is read, it is appended to the previously scanned data in the buffer and the entire buffer is transmitted as one long message.

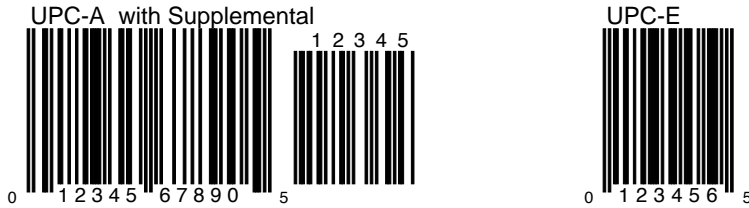
The FULL ASCII version of Code 39 is a modification of the NORMAL (standard) version that can encode the complete 128 ASCII character set (including asterisks). The Full ASCII version is implemented by using the four characters: \$/+%. as *shift* characters to change the meanings of the rest of the characters in the Normal Code 39 character set. Because the Full ASCII version uses shift characters in combination with other standard characters to represent data not in the Normal Code 39 character set, each non-standard character requires twice the width of a standard character in a printed symbol.

The table below shows the character combinations used to produce the Full ASCII version of Code 39.

ASCII	CODE 39	ASCII	CODE 39	ASCII	CODE 39	ASCII	CODE 39
NUL	%U	SP	SPACE	@	%V	`	%W
SOH	\$A	!	/A	A	A	a	+A
STX	\$B	"	/B	B	B	b	+B
ETX	\$C	#	/C	C	C	c	+C
EOT	\$D	\$	/D	D	D	d	+D
ENQ	\$E	%	/E	E	E	e	+E
ACK	\$F	&	/F	F	F	f	+F
BEL	\$G	'	/G	G	G	g	+G
BS	\$H	(/H	H	H	h	+H
HT	\$I)	/I	I	I	i	+I
LF	\$J	*	/J	J	J	j	+J
VT	\$K	+	/K	K	K	k	+K
FF	\$L	,	/L	L	L	l	+L
CR	\$M	-	-	M	M	m	+M
SO	\$N	.	.	N	N	n	+N
SI	\$O	/	/O	O	O	o	+O
DLE	\$P	0	0	P	P	p	+P
DC1	\$Q	1	1	Q	Q	q	+Q
DC2	\$R	2	2	R	R	r	+R
DC3	\$S	3	3	S	S	s	+S
DC4	\$T	4	4	T	T	t	+T
NAK	\$U	5	5	U	U	u	+U
SYN	\$V	6	6	V	V	v	+V
ETB	\$W	7	7	W	W	w	+W
CAN	\$X	8	8	X	X	x	+X
EM	\$Y	9	9	Y	Y	y	+Y
SUB	\$Z	:	/Z	Z	Z	z	+Z
ESC	%A	;	%F	[%K	{	%P
FS	%B	<	%G	/	%L	:	%Q
GS	%C	=	%H]	%M	}	%R
RS	%D	>	%I	^	%N	~	%S
US	%E	?	%J	_	%O	DEL	%T,%X,%Y,%Z

Note: Because all of the characters used to implement Full ASCII Code 39 are part of the Normal Code 39 character set, readers that do not support Full ASCII Code 39 will still read Full ASCII Code 39 symbols. The reader will output shifted characters as if they were normal Code 39 characters. For example the following string: **B-CODER=FUN** will be read as: **B-CODER%HFUN** by a reader that only supports Normal Code 39. Instead of converting the Full ASCII encoded characters %H to an equal sign, the reader blindly outputs %H.

UPC-A, UPC-E, and UPC Supplementals



UPC-A is a 12 digit, numeric symbology used in retail applications. UPC-A symbols consist of 11 data digits and one check digit. The first digit is a number system digit that usually represents the type of product being identified. The following 5 digits are a manufacturers code and the next 5 digits are used to identify a specific product.

UPC numbers are assigned to specific products and manufacturers by the Uniform Code Council (UCC). To apply for a UPC number or for more information, you can contact the UCC at 8163 Old Yankee Road, Suite J, Dayton, OH 45458 Tel: 937-435-3870

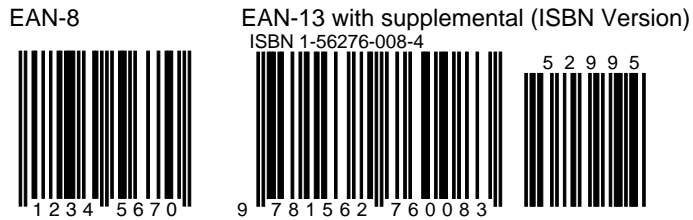
When specifying UPC-A messages, you may enter up to 11 digits. B-Coder will automatically calculate the check digit for you. If you enter less than 11 digits or if you enter any digits other than 0 to 9, B-Coder will prompt you with a warning message. If the option "Enable Invalid Message Warnings" in the Preferences menu is not selected and you do not enter 11 digits, B-Coder will left pad short messages with zeros and truncate longer messages so that the total length is 11.

UPC-E is a smaller, six digit, UPC symbology for number system 0. It is often used for small retail items. UPC-E is also called "zero suppressed" because UPC-E compresses a normal 12 digit UPC-A code into a six digit code by "suppressing" the number system digit, trailing zeros in the manufacturers code and leading zeros in the product identification part of the bar code. A seventh check digit is encoded into a parity pattern for the six main digits. UPC-E can thus be uncompressed into a standard UPC-A 12 digit number.

For UPC-E bar codes, you may enter up to 6 digits. Again, B-Coder will calculate the check digit and truncate or pad the number to a total length of 6 digits as with UPC-A.

Both UPC-A and UPC-E allow for a supplemental two or five digit number to be appended to the main bar code symbol. This supplemental message was designed for use on publications and periodicals. If you enter a supplemental message, it must consist of either two or five numeric digits. The supplemental is simply a small additional bar code that is added onto the right side of a standard UPC symbol.

EAN-8 / EAN-13, BookLand and EAN Supplementals



EAN or European Article Numbering system (also called JAN in Japan) is a European version of UPC. It uses the same size requirements and a similar encoding scheme as for UPC codes.

EAN-8 encodes 8 numeric digits consisting of two country code digits, five data digits and one check digit. B-Coder will accept up to 7 numeric digits for EAN-8. B-Coder will automatically calculate the check digit for you. If you enter less than 7 digits or if you enter any digits other than 0 to 9, B-Coder will display a warning message. If the option "Enable Invalid Message Warnings" in the Preferences menu is not selected and you do not enter 7 digits, B-Coder will left pad short messages with zeros and truncate longer messages so that the total length is 7.

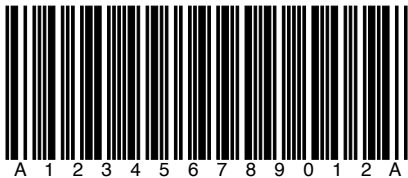
EAN-13 is the European version of UPC-A. The difference between EAN-13 and UPC-A is that EAN-13 encodes a 13th digit into the parity pattern of the left six digits of a UPC-A symbol. This 13th digit, combined with the 12th digit, usually represent a country code.

Both EAN-8 and EAN-13 support a supplemental two or five digit number to be appended to the main bar code symbol. The supplemental is designed for use on publications and periodicals. Supplemental messages must consist of either two or five numeric digits and will appear as a small additional bar code on the right side of a standard EAN symbol.

EAN bar code numbers are assigned to specific products and manufacturers by an organization called **ICOF** located in Brussels, Belgium. Tel: 011-32-2218-7674

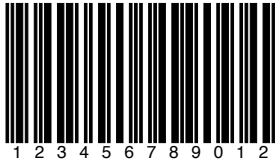
EAN-13 has been adopted as the standard in the publishing industry for encoding ISBN numbers on books. An ISBN or *BookLand* bar code is simply an EAN-13 symbol consisting of the ISBN number preceded by the digits **978**. The supplemental in an ISBN bar code is simply the retail price of the book preceded by the digit **5**. For example, if your ISBN number is 1-56276-008-4 and the price of the book is \$29.95 then you would enter **978156276008** as the bar code message and **52995** for the supplemental. If you choose BookLand as the bar code symbology, B-Coder will automatically create the correct EAN-13 bar code if you supply just the first 10 digits of the ISBN number and optionally a price (without the preceding 5).

CODABAR



CodaBar is a variable length symbology that allows encoding of the following 20 characters: 0123456789-\$./+ABCD. CodaBar is commonly used in libraries, blood banks, and the air parcel business. CodaBar uses the characters A B C and D only as start and stop characters. Thus, the first and last digits of a CodaBar message must be A B C or D and the body of the message should not contain these characters. B-Coder will allow any length of CodaBar message as long as it contains valid characters and starts and ends with a valid start/stop character. If you use lower case letters for A B C or D, B-Coder will convert to upper case.

INTERLEAVED 2 OF 5



Interleaved 2 of 5 is a high density variable length numeric only symbology that encodes digit pairs in an interleaved manner. The odd position digits are encoded in the bars and the even position digits are encoded in the spaces. Because of this, I 2 of 5 bar codes must consist of an even number of digits. Also, because partial scans of I 2 of 5 bar codes have a slight chance of being decoded as a valid (but shorter) bar code, readers are usually set to read a fixed (even) number of digits when reading I 2 of 5 symbols. The number of digits are usually pre-defined for a particular application and all readers used in the application are programmed to only accept I 2 of 5 bar codes of the chosen length. Shorter data can be left padded with zeros to fit the proper length. B-Coder will only accept numeric digits for I 2 of 5 bar codes. If an odd number of digits is entered, B-Coder will Left-Pad one zero to the number entered.

Interleaved 2 of 5 optionally allows for a weighted modulo 10 check character for special situations where data security is important. To enable the I 2 of 5 check character feature in B-Coder, select the option **INCLUDE CODE 39 / I 2 of 5 CHECK CHARACTER** in B-Coder's Preferences menu. When this option is enabled, B-Coder will automatically calculate and append the proper check character to all Interleaved 2 of 5 symbols.

DISCRETE 2 OF 5



Discrete 2 of 5 is a variable length numeric symbology very similar to Interleaved 2 of 5 except that instead of encoding data in both the bars and the spaces, data is only encoded in the bars. Because of this, discrete 2 of 5 is not as compact as Interleaved 2 of 5 and also, odd numbers of digits may be encoded. Use of Discrete 2 of 5 is not very common and few bar code readers support this symbology.

CODE 93



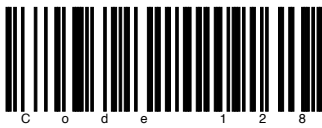
CODE 93 is a variable length symbology that can encode the complete 128 ASCII character set. Code 93 was developed as an enhancement to the CODE 39 symbology by providing a slightly higher character density than CODE 39. CODE 93 also incorporates two check digits as an added measure of security. Although CODE 93 is considered more robust than CODE 39, it has never achieved the same popularity. CODE 93 bar codes are framed by a special start/stop character. B-Coder will automatically add the start and stop characters as well as the check digits to each Code 93 bar code therefore you should not attempt to include them as part of your bar code message.

MSI/Plessey



MSI Plessey is numeric only symbology used primarily for marking grocery store shelves in inventory applications. Because MSI Plessey is not as efficient as most newer symbologies like Code 128 or Interleaved 2 of 5, it is not normally chosen when designing a new bar code application. Each MSI Plessey bar code symbol contains a start code, a variable length data message, a check character and a stop code. B-Coder will automatically insert the start and stop codes as well as automatically calculate and add the required check digit.

CODE 128



Code 128 is a variable length, high density, alphanumeric symbology. Code 128 has 106 different bar and space patterns and each pattern can have one of three different meanings, depending on which of three different character sets is employed. Special *start characters* tell the reader which of the character sets is initially being used and three special *shift codes* permit changing character sets inside a symbol. One character sets encodes all upper case and ASCII control characters, another encodes all upper and lower case characters and the third set encodes numeric digit pairs 00 through 99. This third character set effectively doubles the code density when printing numeric data. Code 128 also employs a check digit for data security. In addition to ASCII characters, Code 128 also allows encoding of four special function codes (FNC1 - FNC4). The meaning of function code FNC1 and FNC4 were originally left open for application specific purposes. Recently an agreement was made by the Automatic Identification Manufacturers Assoc. (AIM) and the European Article Numbering Assoc. (EAN) to reserve FNC1 for use in EAN applications. FNC4 remains available for use in closed system applications. FNC2 is used to instruct a bar code reader to concatenate the message in a bar code symbol with the message in the next symbol. FNC3 is used to instruct a bar code reader to perform a reset. When FNC3 is encoded anywhere in a symbol, any data also contained in the symbol is discarded. The four function codes can be added to a message by selecting them from the bottom of the ASCII chart in B-Coder's window.

Note: B-Coder will automatically select the proper character sets and insert the necessary start character and shift codes so that the resulting bar code will be as short as possible. The check digit will also be calculated automatically by B-Coder.

EAN/UCC 128



The EAN/UCC 128 symbology is a variation of the original Code 128 symbology designed primarily for use in product identification applications. The EAN/UCC 128 specification uses the same code set as Code 128 except that it does not allow function codes FNC2-FNC4 to be used in a symbol and FNC1 is used as part of the start code in the symbol. The check digit in EAN/UCC128 symbols is also calculated differently than in Code 128.

POSTNET



POSTNET (**POSTal N**umeric **E**ncoding **T**echnique) is a 5, 9 or 11 digit numeric only bar code symbology used by the US Postal Service to encode ZIP Code information for automatic mail sorting by zip code. The bar code may represent a five digit ZIP Code (32 bars), a nine digit ZIP + 4 code (52 bars) or an eleven digit Delivery Point code (62 bars).

POSTNET is unlike other bar codes because data is encoded in the height of the bars instead of in the widths of the bars and spaces. Most standard bar code readers cannot decode POSTNET. This symbology was chosen by the Postal Service mainly because it is extremely easy to print on almost any type of printer. POSTNET is a fixed dimension symbology meaning that the height, width and spacing of all bars must fit within exact tolerances.

B-Coder will only create POSTNET bar codes that follow the guidelines published by the Postal Service. B-Coder does not allow direct control over the size of POSTNET bar codes.

NOTE: Most other Windows programs allow you to modify the size of metafile graphic images pasted from the clipboard. If you change the size of a POSTNET bar code even by a small amount, it will be completely unreadable by the Postal Service bar code readers.

B-Coder will ignore non-numeric data in any bar code message that you enter for POSTNET. For example, if you enter "Chicago, IL 60601-3222" for a POSTNET bar code message, B-Coder will still create a correct bar code. This feature allows you to cut an address line from another Windows program and paste it into B-Coder without having to edit the bar code text.

Postal FIM Patterns

FIM or Facing Identification Mark patterns are another type of postal bar code used in automated mail processing by the US Postal Service. FIM patterns are used for automatic facing and canceling of mail that does not contain a stamp or meter imprint (business reply mail, penalty mail, etc.). They also provide a means of separating business and courtesy reply mail from other letters. Three FIM patterns are currently in use. FIM-A is used on courtesy reply mail that has been preprinted with Postnet bar codes. FIM-B is used on business reply, penalty and franked (government) mail that is not preprinted with Postnet bar codes. FIM-C is used on business reply, penalty and franked mail that has been preprinted with Postnet bar codes. FIM patterns are placed in the upper right corner along the top edge and two inches in from the right edge of letters and cards. For more information about all postal bar codes contact your local post office.

When you select a FIM pattern from the symbology menu, B-Coder will immediately generate the requested pattern without switching from the currently selected symbology and without having to select Build/Copy from B-Coder's Edit menu.

PDF417



PDF417 is a high density 2 dimensional bar code symbology that essentially consists of a stacked set of smaller bar codes. The symbology is capable of encoding the entire (255 character) ASCII set. PDF stands for "Portable Data File" because it can encode as many as 2725 data characters in a single bar code. The complete specification for PDF417 provides many encoding options including data compaction options, error detection and correction options, and variable size and aspect ratio symbols. The symbology was published by Symbol Technologies, Inc. to fulfill the need for higher density bar codes. The low level structure of a PDF417 symbol consists of an array of *code words* (small bar and space patterns) that are grouped together and stacked on top of each other to produce the complete printed symbol. An individual code word consists of a bar and space pattern 17 *modules* wide. The user may specify the module width, the module height, and the overall aspect ratio (overall height to width ratio) for the complete symbol. A complete PDF417 symbol consists of at least 3 rows of up to 30 code words and may contain up to 90 code word rows per symbol with a maximum of 928 code words per symbol. Shown below is the basic structure of a typical PDF417 bar code.

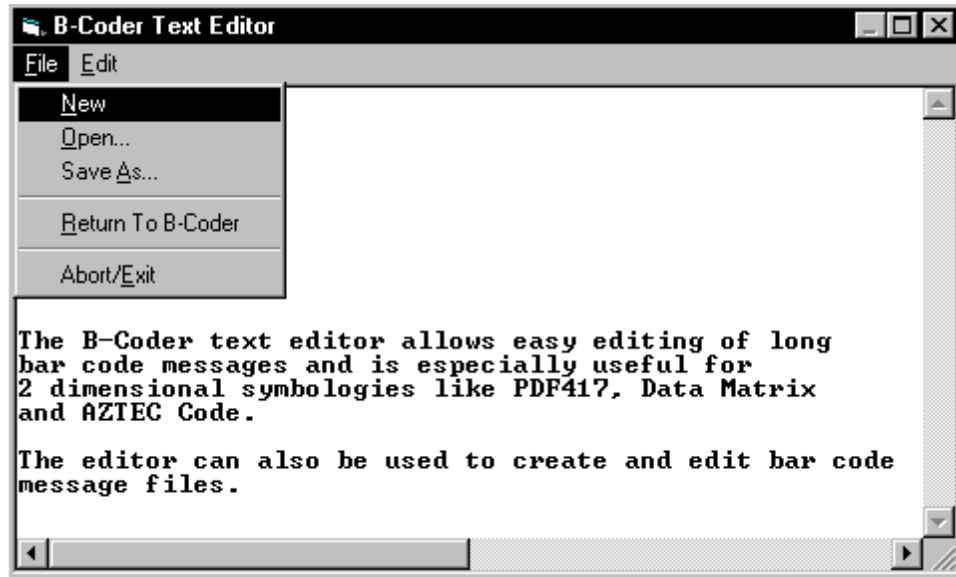


The code words in a PDF417 symbol are generated using one of three data compaction modes currently defined in the symbology specifications. This allows more than one character to be encoded into a single data code word. Because different data compaction algorithms may be used, it is possible for different printed symbols to be created from the same input data. The symbology also allows for varying degrees of data security or error correction and detection. Nine different error correction levels are available with each higher level adding additional overhead to the printed symbol.

B-Coder allows complete control over all optional features of PDF417 by selecting **PDF417 Options...** from the Preferences menu.

The PDF417 Text Editor

When you select PDF417 from the symbologies menu, the toolbar in B-Coder's Window will display a button with the caption **Editor**. If you click your mouse on this button, a text editor will appear where you can more easily edit long bar code messages. The editor functions much like the Notepad program that comes with Windows except that it is built into B-Coder. When you open the text editor, the Bar Code Message text is copied into the editor and when you finish editing the text, it will be replaced in the bar code message text field in B-Coder's window. Shown below is the B-Coder PDF417 text editor.



The **File** menu in the text editor allows you to open and save disk files as well as return to B-Coder. When you are finished editing a bar code message, you would normally select **Return To B-Coder** from the editor's file menu. This option will quit the editor and update the Bar Code Message text field in B-Coder's main window with the edited text.

The option: **Abort/Exit** will return you to B-Coder's main window without updating the bar code message text field.

The **Edit** menu in the text editor provides the three editing commands **Cut**, **Copy** and **Paste**.

Note: The text editor is designed for editing text messages only and should not be used for editing or inputting binary data. (i.e. files containing characters that cannot be typed in)

PDF417 Data Compaction Modes

At the time of this writing, there are three pre-established data compaction modes for the PDF417 symbology. The three modes are listed below:

1. **Extended Alphanumeric Compaction (EXC)** mode. This mode supports encoding of all printable ASCII characters and can compress approximately 2 characters per code word.
2. **Binary/ASCII Plus** mode. *Binary* mode supports encoding of the entire ASCII character set. Binary mode can compress approximately 1.2 characters per code word.
3. **Numeric Compaction** mode. *Numeric* mode can encode only the numeric characters 0 through 9 and can achieve data compression of approx. 3 characters per code word.

One or more modes can be used to generate a PDF417 symbol by using special *shift* and *latch* code words to switch between modes within the symbol. B-Coder gives you the option to select any of the three standard data compaction modes as well as two **Auto** modes that automatically generate the smallest number of code words (and thus the smallest symbol) by intelligently switching between the three standard modes.

The first of the 2 *Auto* modes provided by B-Coder is called **Auto (EXC/Binary/Numeric)**. This mode allows full switching between the three standard data compaction modes and provides the maximum data compression possible. The second mode, **Auto (EXC/Binary)**, is similar to the first except that it does not allow switching to Numeric mode in a symbol. The only reason for this mode is to provide support for a small number of early PDF417 readers that did not have the capability to decode Numeric mode encoded data..

PDF417 Error Detection and Correction

One of PDF417's primary features is error detection and correction or *data security*. Each PDF417 symbol has 2 code words for error detection. (Similar to a check digit in standard bar code symbols.) The error correction capacity may be selected by the user, based on the needs of a specific application. Error correction compensates for label defects and misdecodes. There are essentially 2 types of errors that can occur in a bar code symbol, **Erasures** and **Misdecodes**. Erasures are missing or undecodable code words and misdecodes are errors that cause the reader to interpret a particular code word incorrectly. Nine error correction levels numbered 0 through 8 are available in the symbol specification. Each higher security level allows for a higher number of erasures and misdecodes to be recovered from. (Since it requires 2 code words to recover from a misdecode, one to detect the error and one to correct for it, a given security level can support half the number of misdecodes that it can undecoded or missing code words.)

Since error correction and detection is encoded into additional code words, each higher security level adds additional overhead to a printed symbol. Higher security levels reduce the maximum number of data characters that can be encoded in a symbol and also increase the size of a printed symbol. (A single PDF417 symbol can contain a maximum of 928 total code words for data and error correction combined.)

The relationship between security level, error correction capacity and the number of additional code words or *overhead* required for a given security level is as follows:

Error Correction Level	Maximum Limit of Allowable Erasures + (2 x Misdecodes)	Symbol Overhead (Number of Additional Code words)
0	0	2
1	2	4
2	6	8
3	14	16
4	30	32
5	62	64
6	126	128
7	254	256
8	510	512

Truncated PDF417 Symbols

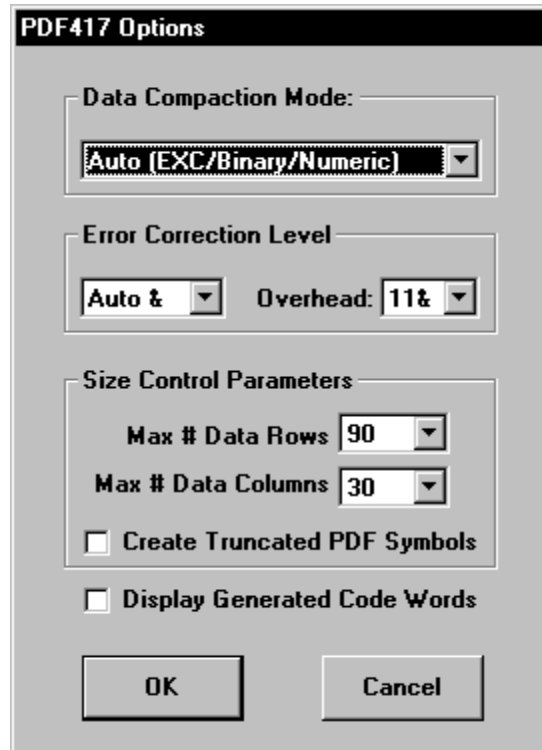
In an effort to further reduce the size of a PDF417 symbol, the designers of the symbology allowed for a variation of PDF417 called **Truncated PDF417**. A truncated PDF417 symbol is almost identical to a standard PDF417 symbol except that the right row indicator code words and the right hand stop pattern are omitted and replaced with a narrow bar one module wide.

Truncated PDF417 symbol:



The B-Coder PDF417 Options Dialog Box.

B-Coder allows for a great deal of control over how PDF417 symbols are generated including full selection of the data compression scheme, the security level, and several additional controls over the size and aspect ratio of a generated symbol. When you select **PDF417 Options...** from the **Preferences** menu, the following dialog box will appear:



The dialog box is titled "PDF417 Options". It contains three main sections:

- Data Compaction Mode:** A dropdown menu currently showing "Auto (EXC/Binary/Numeric)".
- Error Correction Level:** A dropdown menu showing "Auto %" and an "Overhead:" dropdown menu showing "11%".
- Size Control Parameters:** Two dropdown menus: "Max # Data Rows" set to "90" and "Max # Data Columns" set to "30". Below these are two checkboxes: "Create Truncated PDF Symbols" (unchecked) and "Display Generated Code Words" (unchecked).

At the bottom of the dialog are "OK" and "Cancel" buttons.

The three main sets of options available are **Data Compression Mode**, **Security Level** and **Size Control Parameters**.

The five Data Compression modes available in B-Coder are **AUTO(EXC/Binary/Numeric)**, **AUTO(EXC/Binary)**, **EXC**, **Binary** and **Numeric**. For almost all applications the Data Compression option should be left set to AUTO(EXC/Binary/Numeric). This mode will provide the maximum data compression possible for any given bar code message. The other four data compression options are only provided to allow experimenting with the other compression modes.

The Error Correction Level options allow you to select a specific PDF417 error correction level from 0 to 8. You can also have B-Coder automatically select an error correction level based on a percentage of total symbol area to be used for error correction. If you select **Auto%** for the Error Correction Level, a list box will appear where you can choose the percentage of error correction overhead to add from 0% to 99%. The Auto% Error Correction level mode is probably the best way to specify a security level because it guarantees that you will not waste symbol real estate with more error correction overhead than is necessary for small messages. It also insures that enough error correction will be generated for larger messages. (A small message only needs a small amount of error correction capacity while a larger message needs more.) The default percentage of Overhead when Auto% is selected is 11%.

The additional Size Control Parameters that may be specified are the **Maximum Number of Data Rows and Columns** as well as an option to enable printing of **Truncated PDF417** symbols. The default maximum number of data rows and columns are the maximum values allowable in the PDF417 symbol specification. If you specify smaller values for these parameters, you are essentially defining an upper limit to the overall height or width of all generated PDF symbols. For example, suppose you need to generate a PDF symbol that absolutely must be no wider than two inches and you chose 10 mils for the module width. Because there are 17 modules in a code word and there are 69 modules of overhead per row, (start/stop patterns and right/left row indicators) the maximum number of code word columns allowable in a 2 inch wide symbol can be calculated using the formula:

$$10\text{mils} \times [(17 \times \text{MaxCols}) + 69] \leq 2000 \text{ mils (2 inches)}$$

If we solve the equation above we obtain the value of 7 for the Maximum number of code word columns (MaxCols). Thus, if we set the Maximum Number of Data Columns parameter to 7, B-Coder will only allow you to produce PDF417 symbols that are less than 2 inches wide.

The remaining size control parameter, **Create Truncated PDF417 Symbols**, causes B-Coder to create the truncated version PDF417 that removes the right row indicator and the right hand stop pattern from the symbol. When creating Truncated PDF417 symbols, there are only 35 overhead modules per row instead of the normal 69 overhead modules.

The final option in the PDF417 Options dialog box, **Display Generated Code words**, causes B-Coder to display a list of all the code word values generated for each PDF417 symbol. The code word list will appear directly below the bar code message text in B-Coder's window. This option was included for use by PDF417 bar code reader system developers to see exactly data code words are being generated by B-Coder. Note: The first code word in every PDF417 bar code is a count of the total number of data code words contained in the symbol.

BPO 4 State Code



BPO (British Post Office) 4 State Code is a new postal bar code symbology that has been developed by the British Post office for encoding European postcode data similar to the way the US POSTNet symbology is used for encoding Zip Code data. The goal of BPO 4 State Code is to provide European countries with a simple and efficient postal bar coding scheme.

The U.S. POSTNet symbology encodes numeric characters in a pattern of four bars per character with each bar being either tall or short (i.e. two possible “states” for each bar). The U.S. technique thus allows for up to 16 different possible bar patterns for each set of four bars and is adequate for encoding the ten digits zero through nine. Because European postcodes contain both alpha and numeric characters, (thus requiring a minimum of 36 different possible patterns for the characters A-Z and 0 to 9), each character in the BPO 4 State Code is encoded into four bars with each bar having *four* possible “states”. The four states are: tall bars, short bars, medium height bars extended up from the middle of the symbol and medium height bars extended down from the middle of the symbol. In theory, the BPO 4 State Code is capable of encoding up to 128 different characters however only the characters A through Z and 0 to 9 have been assigned unique bar patterns.

BPO 4 State Code is a fixed dimension symbology meaning that the height, width and spacing of all bars must fit within exact tolerances. B-Coder will only create BPO 4 State Code bar codes that follow the guidelines published by the British Postal Service. B-Coder does not allow direct control over the size of BPO 4 State Code bar codes.

NOTE: Most other Windows programs allow you to modify the size of metafile graphic images pasted from the clipboard. If you change the size of a BPO 4 State Code bar code even by a small amount, it will be completely unreadable by European Postal Service bar code readers.

RSS14



RSS14 (**R**educed **S**pace **S**ymbology) is a relatively new symbology that encodes up to 14 numeric digits (13 digits plus a check digit) along with an optional two dimensional “Composite” bar code component that can encode up to 338 additional bytes of alpha/numeric data. The principle use of the RSS symbology is to identify items that cannot be marked with current linear symbols because of size restrictions. The Composite Symbols provide additional supply chain data while allowing for the co-existence of symbologies already being used.

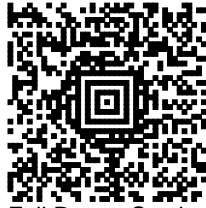
For more information on the RSS14 symbology, visit: <http://www.uc-council.org>

The recommended Bar Height for standard RSS bar code symbols is .43 inches and the recommended Narrow Bar Width is 13 mils.

Aztec Code



Compact Symbol



Full Range Symbol

Aztec Code is a high density 2 dimensional matrix style bar code symbology that can encode up to 3750 characters from the entire 256 byte ASCII character set. The symbol is built on a square grid with a bullseye pattern at its center. Data is encoded in a series of "layers" that circle around the bullseye pattern. Each additional layer completely surrounds the previous layer thus causing the symbol to grow in size as more data is encoded yet the symbol remains square. Aztec's primary features include: a wide range of sizes allowing both small and large messages to be encoded, orientation independent scanning and a user selectable error correction mechanism.

The smallest element in an Aztec symbol is called a "module" (i.e. a square dot). The module size and the amount of error correction are the only "dimensions" that can be specified for an Aztec symbol and both are user selectable. It is recommended that the module size should range between 15 to 30 mils in order to be readable by most of the scanners that are currently available.

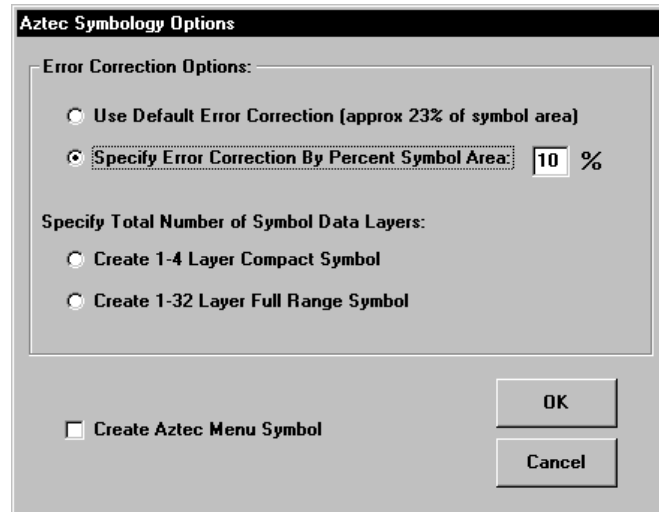
The overall size of an Aztec symbol is dependent on the module size, the total amount of encoded data and also on the level of error correction capacity chosen by the user. The smallest Aztec symbol is 15 modules square and can encode up to 14 digits with 40% error correction. The largest symbol is 151 modules square and can encode 3000 characters or 3750 numeric digits with 25% error correction.

There are three types of Aztec symbols, a "Compact" symbol, a "Full Range" symbol and a "Menu" symbol. Compact symbols have a smaller bullseye pattern and are limited in their overall size to having up to four "layers" of data surrounding the bullseye pattern. Full Range symbols have a larger bullseye pattern and can have up to 32 layers of data surrounding the bullseye. Do not confuse "Layers" with modules. A layer actually is made up of two stacked modules resembling a domino with each domino laid out around the bullseye pattern so that the long edge of the domino points away from the center of the symbol.

Menu symbols are a special type of Aztec bar code that are typically used by scanner manufacturers to create bar codes that contain commands for enabling and disabling features in a bar code reader. Menu symbols can be either Compact or Full Range symbols and are only useful if you know the commands that a particular reader will recognize.

AZTEC Code Advanced Options

B-Coder allows you to choose the module size for your Aztec symbols directly in B-Coder's main window. You can specify the amount of error correction by selecting "Aztec Code Advanced Options..." from the Preferences menu. When you do so, the following dialog box will appear allowing you to specify the amount of error correction in one of three ways:



1. You may choose the default error correction which will consume roughly 23% of the total symbol area.
2. You can specify the percentage of the total symbol area that you would like to reserve for error correction. In this case you can enter a percentage value between 1% and 99%.
3. You can specify the total number of data layers that you want the symbol to contain. This method of specifying error correction has two sub-options where you can tell B-Coder to generate either Compact symbols with 1 to 4 layers or Full Range symbols with 1 to 32 layers. Because you are specifying the total number of data layers, all of your Aztec symbols will be the same size. If all of the layers in the symbol are not used for data, any remaining empty layers will be used for error correction. If the message that you are trying to encode requires more layers than the number of layers that you specify, B-Coder will display an error message.

For most normal applications, you should use the default error correction.

Data Matrix



Data Matrix is a high density 2 dimensional matrix style bar code symbology that can encode up to 3116 characters from the entire 256 byte ASCII character set. The symbol is built on a square grid arranged with a finder pattern around the perimeter of the bar code symbol.

There are two types of Data Matrix symbols each using a different error checking and correction scheme (ECC). The different types of Data Matrix symbols are identified using the terminology "ECC" followed by a number representing the type of error correction that is used by the encoding software. ECC 000 to ECC 140 are the original type of Data Matrix symbols and are now considered obsolete. The newest version of Data Matrix is called ECC 200 and is recommended for all new Data Matrix applications. The ECC 200 version of Data Matrix uses a much more efficient algorithm for encoding data in a symbol as well as an advanced error checking and correction scheme. B-Coder fully supports all variations of the Data Matrix symbology however the author of the original symbology specification (CI Matrix Co.) highly recommends that ECC 000 - ECC 140 be used only where absolutely necessary.

ECC 000 - ECC 140

Data Matrix symbols designated by the terms ECC 000 to ECC 140 are the original type of Data Matrix symbol that uses a convolutional error correction scheme. There are actually five levels of error correction available for this type of Data Matrix symbol with each higher level of error correction designated as follows.

- ECC 000 - Provides no error correction
- ECC 050 - Provides error correction for damage of up to 2.8% of the printed symbol.
- ECC 080 - Provides error correction for damage of up to 5.5 % of the printed symbol.
- ECC 100 - Provides error correction for damage of up to 12.6% of the printed symbol.
- ECC 140. - Provides error correction for damage of up to 25% of the printed symbol.

For all five ECC levels ECC 000 - ECC 140 there is also a user selectable option for a "Data Format" which defines the type of data that may be encoded in a Data Matrix symbol. The available formats are listed below:

Allowable Data In The Bar Code Message

- Numeric digits 0 to 9 and the space character
- Upper case alpha A-Z and the space character
- Upper case alphanumeric A-Z, 0-9 and the space character
- A-Z, 0-9, space, minus, period, comma & forward slash (/)
- 7 bit ASCII - all ASCII characters between ASCII 0 to ASCII 127
- 8 bit ASCII - all ASCII characters between ASCII 0 to ASCII 255

ECC200

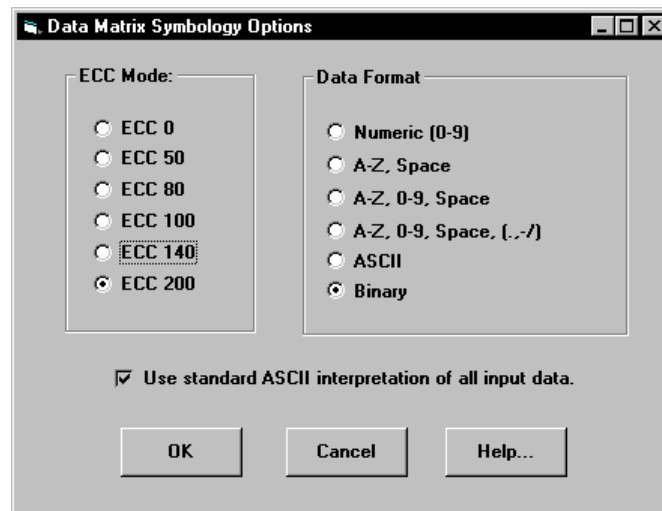
ECC 200 is the latest and most advanced version of Data Matrix and is therefore strongly recommended for use in all new Data Matrix applications. ECC 200 uses a Reed-Solomon error correction algorithm along with a code set switching mechanism that is much more efficient at packing data into a symbol than any of the earlier encoding schemes (ECC 000 - ECC 140). ECC 200 is capable of encoding the entire 8 bit ASCII character set in a highly efficient manner and therefore does not provide for or require any Data Format options. In addition to encoding 8 bit ASCII data, ECC 200 also allows for a special function code called "FNC1" as well as special indicators for features like "Structured Append" and "Extended Channel Interpretation".

Structured append is a means for generating multiple bar codes containing a much larger data message than can be encoded into a single symbol. With structured append, each bar code contains a portion of a larger data message along with a number that identifies the portion of the message contained in the symbol. When a bar code reader scans a symbol that is part of a sequence, it will not transmit the data until all symbols in a sequence have been read. It does not matter what order the bar codes are read in - the reader will correctly build the complete message.

Extended Channel Interpretation is a mechanism for creating user definable data encoding schemes. For example, suppose you wanted to replace the standard ASCII character set with a different character set (a foreign language character set for example), you could use the Extended Channel Interpretation feature in ECC 200 to indicate that the message encoded in a symbol conforms to the new interpretation. Notes: To encode data to conform to specific industry standard it needs to be authorized by AIM International.

Data Matrix Advanced Options

When you select the Data Matrix symbology in B-Coder, a menu item labeled "Advanced Data Matrix Options" will appear in the B-Coder Preferences menu. If you select this menu item, B-Coder will display the following dialog box where you can choose the ECC level and Data Format for your Data Matrix bar code symbols.



The default ECC option is ECC 200 which is the latest revision of the Data Matrix symbology specification. ECC 200 provides the best data compression as well as the best error checking and correction capability of all the available ECC options and is therefore recommended for all new Data Matrix applications.

If you select any ECC option other than ECC 200, then you will also be able to choose a Data Format option that will restrict the type of data that you will be able to encode in your Data Matrix bar codes. The available options are: Numeric, Upper Case Alpha (A-Z and Space), Upper Case AlphaNumeric, Upper Case AlphaNumeric with Punctuation, ASCII and Binary. For example, if you choose Numeric as the data format, you will only be able to encode the numeric characters 0 to 9 in your bar codes.

If you select the ECC 200 option the data format options are not valid and you will be able to encode the complete 256 byte ASCII character set including binary data.

The ECC 200 option also supports several advanced features including special function codes that can be encoded in a bar code as well as indicators for structured append which allows large data messages to be split into multiple bar code symbols.

Because ECC 200 supports the entire 8 bit ASCII character set and therefore cannot use ASCII character values to represent special features (like FNC1), B-Coder provides two methods for interpreting the input data message. B-Coder has an option in the Data Matrix Advanced Options dialog box labeled "Use Standard ASCII Interpretation For All Input Data". This option determines how the input data message for Data Matrix bar codes is interpreted.

If this option is selected, then the bar code message that you enter is assumed to not contain any special function codes like FNC1 or Extended Channel Interpretation codes. For most typical applications this option would normally be used.

If the "Use Standard ASCII Interpretation For All Input Data" option is not checked, then the tilde character (~) can be used in the bar code message as an indicator that the character(s) following the tilde are to be interpreted with a special meaning as outlined below. To encode a tilde (~) use the string: ~~ (i.e. two tilde characters). If no tilde characters or Nuls (ASCII 0) are present in the input message, then enabling the "Use Standard ASCII Interpretation For All Input Data" option has no effect on the resulting bar code symbol.

Tilde Control Codes

~X (a tilde character followed by any upper case alpha character) is used as a shift character for inserting control codes (characters with ASCII values 0 to 26) into a bar code message. For example, ~@ = NUL, ~A = ASCII 1, ~G = BEL (ASCII 7), ~M = ASCII 13 (carriage return). If you need to insert ASCII control codes into a message, take the ASCII value for the control code (1-26) and find the corresponding letter in the alphabet and precede it with a tilde. i.e. The ASCII value for a carriage return character is ASCII 13 and the thirteenth letter of the alphabet is "M" therefore to insert a carriage return in a bar code message, you would use "~M". Note: You can also pass control codes directly to the DLL without having to use the ~ before an alpha character. For example you could use either an ASCII 13 character or the sequence ~M to represent a carriage return.

~1 is used to represent the FNC1 code and is followed by normal data.

To encode data to conform to specific industry standards as authorized by AIM International, a FNC1 character shall appear in the first or second symbol character position (or fifth or sixth data position of the first symbol of structured append). FNC1 encoded in any other position is used as a field separator and shall be transmitted as a GS control character (ASCII value 29).

Notes: To encode data to conform to specific industry standard, it needs to be authorized by AIM International. Contact AIM International at Tel: 703-391-7621 or email: adc@aimi.org

If the FNC1 code is used in the second character position, the input data before '~1' must be, between 'A' and 'Z', or between 'a' and 'z' or 2-digits between '01' and '99'.

~2 : is used to represent Structured Append and must be followed by a three 3-digit number between 1 and 255 representing the symbol sequence as well as a file identifier of six numeric digits. The file identifier is used to uniquely identify a sequence so that only logically linked sequences are processed as part of the same sequence. The symbol sequence identifier is a number between 1 and 255 that indicates the position of the symbol within a sequence of up to 16 symbols. The sequence identifier actually contains two four bit values representing the sequence number and the total number of symbols in the sequence (i.e. m of n where m is the sequence number and n is the total number of symbols). The upper four bits of this value represent the position of the particular symbol as the binary value of (m-1) and the lower order four bits identify the total number of symbols to be concatenated as the binary value of (17-n). For example, symbol 3 in a sequence of 7 symbols with file ID: 001015 is represented by ~2042001015. The number 042 is derived as follows: $3-1=2$ which equals 0010 when represented as a 4 bit binary number. $17-7=10$ which equals 1010 when represented as a 4 bit binary number. After concatenating the two 4 bit binary values we end up with 00101010 which equals 42 in decimal.

~3 : Indicates that a message is to be used for reader programming purposes and is followed by normal data. This feature is only useful if you know the specific programming commands for your bar code reader.

~5 and **~6** : indicates that the data will contain an abbreviated format header and trailer followed by normal data. The ~5 or ~6 must appear as the first two characters in a message and must not be used in conjunction with structured append. Data Matrix provides a means of abbreviating an industry specific header and trailer in one symbol character. This feature exists to reduce the number of characters needed to encode data using certain structured formats. If a ~5 is used as the first two characters of a message, the header **[]>** + ASCII 30 + **05** + ASCII 29 will be transmitted by the reader before the data in the message and the trailer ASCII 30 + ASCII 4 will be transmitted following the data. Likewise, if a ~6 is used as the first two characters of a message, the header **[]>**+ ASCII 30 + **06** + ASCII 29 will be transmitted by the reader before the data in the message and the trailer ASCII 30 + ASCII 4 will be transmitted following the data.

~7NNNNNN is used to indicate Extended Channel NNNNNN where NNNNNN is 6-digit EC value (000000 - 999999). e.g. Extended Channel 9 is represented by ~7000009

~dNNN creates ASCII decimal value NNN for a codeword (must be 3 digits). Please refer to the official Data Matrix symbology specification for details on the meanings of all codeword values for ECC 200. Contact AIM International at Tel: 703-391-7621 or email: adc@aimi.org

Understanding Dynamic Data Exchange

Dynamic Data Exchange (DDE) is a way for two Windows programs to share data or send commands to each other. DDE can be thought of as a direct communications link between two application programs. In most cases one program is providing some form of data (either text or graphics) to another program. The application that is the source of the data is called the *server* and the application that is receiving the data is called the *client*. Thus, B-Coder is a server application that "serves" bar codes to other (client) programs through the clipboard.

A powerful feature of Dynamic Data Exchange is that a client can send commands to a server to force it to perform specific functions. The types of commands that a server program can accept will vary depending on the server. B-Coder for example understands several dozen different DDE commands that allow other Windows programs to take complete control of it. In fact, anything that you can do from B-Coder's main menu can be done under the control of another Windows program using B-Coder's DDE command language. For example, you could write a program in a word processor's macro language that fully automates the process of generating and retrieving bar codes from B-Coder. B-Coder can even be shrunk down to an icon and still process all DDE commands sent to it.

The communication process between a client and a server is similar to a telephone call. The conversation is always initiated by the client application. A client can usually initiate a DDE link using some form of **DDEInitiate** command in its macro or programming language. The *Initiate* command is essentially a query containing a **DDE Application Name**, and a **DDE Topic** to all other applications currently running. The Application Name, and Topic are analogous to a telephone number and serve to identify the application that the client wants to talk to.

If a server application is running that recognizes the specified DDE Application Name and DDE Topic, and if it responds to the DDE initiate, the Windows operating system opens a "link" or communications channel between the two programs. This process is actually called "Establishing a Link". Once a link has been established, the client can send commands to the server using some form of **Execute** or **DDEexecute** macro command.

When the client no longer needs to communicate with the server it can terminate the DDE link using a **Terminate** or **DDETerminate** macro command. Ending either of the two linked applications will also automatically terminate all DDE links between the two programs.

DDE Commands Supported By B-Coder

B-Coder supports a complete set of DDE commands that can be issued by other Windows programs allowing them to completely take control over all B-Coder functions. When you initiate a DDE link to B-Coder use the DDE Application Name **B-Coder** and the DDE Topic **SYSTEM**. (No DDE Item is necessary). The following is a list of all DDE commands that can be issued to B-Coder. Note: All DDE commands must be enclosed in square brackets.

Command	Action Performed
[APPEXIT]	Removes B-Coder from memory.
[APPMINIMIZE]	Minimizes the B-Coder window to an Icon.
[APPNORMAL]	Opens the B-Coder window to normal size.
[BEEP]	Causes B-Coder to beep one short beep.
[NEW]	Resets B-Coder to its default settings.
[OPENCONFIG(<i>filename</i>)]	Opens a B-Coder configuration file with the specified <i>filename</i> .
[SAVECONFIG(<i>filename</i>)]	Saves the current configuration to disk with the specified <i>filename</i> .
[SAVECONFIG]	Saves the current configuration to disk with the most recently used configuration file name.
[READMSGFILE(<i>filename</i>)]	Reads in the contents of the specified file as a bar code message.
[OPENMSGFILE(<i>filename</i>)]	Opens the specified file containing multiple bar code messages.
[CLOSEMSGFILE]	Closes any open bar code message file.
[SAVEBARCODE(<i>filename</i>)]	Saves the current bar code symbol to disk with the specified <i>filename</i> .
[SAVEBARCODE]	Saves the current bar code to disk with the most recently specified filename.
[BUILD/COPY]	Generates a bar code and copies it to clipboard.
[PASTE]	Pastes text from the clipboard into the bar code message
[PASTE/BUILD/COPY]	Pastes text into the bar code message and generates a bar code
[BARCODE= <i>text</i>]	Generates a bar code using the supplied text and copies it to clipboard. The <i>text</i> argument specifies the bar code message.
[PASTECOMMENT]	Pastes text from the clipboard into the comment line
[COPYMESSAGE]	Copies the Bar Code Message text to the clipboard
[COPYCOMMENT]	Copies the Comment Line text to the clipboard
[MESSAGE= ' <i>text</i> ', <i>n</i> ,...]	Changes the bar code message to the supplied <i>text</i> or ASCII values. The text must conform to the rules for the chosen symbology.
Note:Text must be supplied as quoted strings using single quotes and control codes must be supplied as ASCII values. Text and control codes may be combined with individual elements separated by commas. Ex. [Message='Test',13,10] changes the bar code message to the word <i>Test</i> followed by a carriage return (ASCII 13),and a line feed (ASCII 10).	
[SUPPLEMENT= ' <i>text</i> ']	Changes UPC/EAN supplemental to the supplied <i>text</i> .
Note:Text must consist of either a two or five digit number and must be supplied as a quoted string using single quotes. Ex. [Supplement='12345'] changes the UPC/EAN supplemental message to <i>12345</i> .	
[COMMENT= <i>text</i>]	Changes comment text to the supplied <i>text</i> .
Note: Unlike the MESSAGE= and SUPPLEMENT= commands, text must not be supplied in quotes	
[COMPOSITE= <i>text</i>]	Changes RSS14 Composite message to the supplied <i>text</i> .
Note: Unlike the MESSAGE= and SUPPLEMENT= commands, text must not be supplied in quotes.	
[BARWIDTH= <i>n</i>]	Changes the value for the narrow bar width to <i>n</i> .
[BARHEIGHT= <i>n</i>]	Changes the value for the bar code height to <i>n</i> .
[MAGNIFY= <i>n</i>]	Changes the UPC/EAN magnification factor to <i>n</i>
[RATIO= <i>n</i>]	Changes the Narrow To Wide Element Ratio to <i>n</i>
[CLIPFORMAT=WMF]	Sets the clipboard format to Windows Metafile.
[CLIPFORMAT=BMP]	Sets the clipboard format to Windows Bitmap.
[CLIPFORMAT=DIB]	Sets the clipboard format to Device Independent Bitmap.
[CLIPFORMAT=ALL]	Sets the clipboard format to all of the above.

[CODE39]	Selects Code 39 as default symbology.
[CODE39FULLASCII]	Selects Code 39 Full ASCII as default symbology.
[CODABAR]	Selects CodaBar as default symbology.
[CODE93]	Selects Code 93 as default symbology.
[CODE128]	Selects Code 128 as default symbology.
[UPCA]	Selects UPC-A as default symbology.
[UPCE]	Selects UPC-E as default symbology.
[UPCE1]	Selects UPC-E1 as default symbology.
[EAN8]	Selects EAN-8 as default symbology.
[EAN13]	Selects EAN-13 as default symbology.
[I2OF5]	Selects Interleaved 2 of 5 as default symbology.
[D2OF5]	Selects Discrete 2 of 5 as default symbology.
[EAN/UCC128]	Selects EAN/UCC 128 as default symbology.
[PDF417]	Selects PDF417 as default symbology.
[POSTNET], [ZIP]	Selects POSTNET as default symbology.
[BPO]	Selects BPO 4 State Code as default symbology.
[RSS14], [RSS]	Selects RSS14 as the default symbology.
[AZTEC]	Selects Aztec Code as the default symbology
[DATAMATRIX]	Selects Data Matrix as the default symbology
[MSI]	Selects MSI/Plessey as the default symbology
[AZTECECP= n]	Sets the Aztec Error Checking and Correction value. The allowable values for n are: 0 = Default error correction of approx 23% of symbol area 1-99 = percentage of symbol area to reserve for error correction 101-104 = Compact Aztec symbol with 1 to 4 layers 201-232 = Full Range Aztec symbol with 1 to 32 layers
[AZTECMODULESIZE=n]	Sets the size of the modules in an Aztec symbol.
[AZTECMENU=ON]	Enables generation of Aztec menu symbols.
[AZTECMENU=OFF]	Disables generation of Aztec menu symbols.
[DMMODULESIZE= n]	Sets the size of the modules in a Data Matrix symbol
[DMECC=n]	Sets the Error Checking and Correction value for Data Matrix The allowable values for n are: 0, 50, 80, 100, 140 and 200
[DMFORMAT=n]	Sets the Data Matrix format for ECC 0 - ECC 140 The allowable values for n are: 1 = Numeric digits 0 to 9 and the space character 2 = Upper case alpha A-Z and the space character 3 = Upper case alphanumeric A-Z, 0-9 and the space character 4 = A-Z, 0-9, space, minus, period, comma & forward slash (/) 5 = 7 bit ASCII - all ASCII characters between ASCII 0 to ASCII 127 6 = 8 bit ASCII - all ASCII characters between ASCII 0 to ASCII 255
[DMSTANDARD=ON]	Treat input data to a Data Matrix bar code as standard ASCII data.
[DMSTANDARD=OFF]	Instructs B-Coder to scan input data to a Data Matrix bar code for special tilde codes and replace them with their equivalent data. See pg. 48 for a complete description of all Data Matrix tilde codes.
[INCPREAMBLE=text]	Sets the Message Preamble to <i>text</i> for Incrementally Numbered bar code messages.
[INCPREAMBLE=text]	Sets the Message Postamble to <i>text</i> for Incrementally Numbered bar code messages.
[INCSTART=n]	Sets the starting number to <i>n</i> for Incrementally Numbered bar code messages.
[INCVALUE=n]	Sets the Increment Value to <i>n</i> for Incrementally Numbered bar codes.
[INCWIDTH=n]	Sets the width of the incremented number for incrementally numbered bar codes.
[INCENABLE]	Enables Incremental Numbering.
[INCDISABLE]	Disables Incremental Numbering.
[INCMESSAGE]	Applies incremental numbering to the bar code message.
[INCSUPPLEMENT]	Applies incremental numbering to the UPC/EAN supplemental message.
[NEXT]	Advances to the next message in a series when incremental numbering is enabled or when a bar code message file is open.
[PREV], [PREVIOUS]	Backs up one message in a series when incremental numbering is enabled or when a bar code message file is open.

[TEXT=ON]	Enables human readable text in bar codes.
[TEXT=OFF]	Disables human readable text in bar codes.
[TEXTABOVE=ON]	Prints human readable text above all bar codes.
[TEXTABOVE=OFF]	Prints human readable text below bar codes.
[COMMENTABOVE=ON]	Prints comment text above all bar codes.
[COMMENTABOVE=OFF]	Prints comment text below all bar codes.
[QUIETZONES=ON]	Enables automatic Quiet Zones around all bar codes.
[QUIETZONES=OFF]	Disables automatic Quiet Zones around all bar codes.
[OPTIONALCHECK=ON]	Enables Optional Check Characters for Code 39, I 2 of 5, Codabar and MSI/Plessey bar codes.
[OPTIONALCHECK=OFF]	Disables Optional Check Characters for Code 39, I 2 of 5, Codabar and MSI/Plessey bar codes.
[BarReduction= <i>n</i> %]	Sets the Bar Width Reduction factor to <i>n</i> %. (-75 to 75 percent)
[HORIZONTAL]	Sets the symbol orientation to horizontal.
[VERTDOWN]	Sets the symbol orientation to Vertical rotated 90 degrees
[VERTUP]	Sets the symbol orientation to Vertical rotated 270 degrees
[MESSAGEWARNINGS=ON]	Enables invalid message warnings.
[MESSAGEWARNINGS=OFF]	Disables invalid message warnings.
[PRINTWARNINGS=ON]	Enables printer resolution warnings.
[PRINTWARNINGS=OFF]	Disables printer resolution warnings.
[BEARERBARS=ON]	Enables printing of Bearer Bars with a symbol.
[BEARERBARS=OFF]	Disables printing of Bearer Bars with a symbol.
[INCHES]	Sets the measurement units to Inches/Mils
[METRIC]	Sets the measurement units to Metric
[DOTS]	Sets the measurement units to Printer Dots
[FONTNAME= <i>font</i>]	Sets the font name for human readable text to <i>font</i> .
[FONTSIZE= <i>n</i>]	Sets the font size to <i>n</i> points.
[FONTSCALING=ON]	Enables Automatic Font Scaling.
[FONTSCALING=OFF]	Disables Automatic Font Scaling.
[FONTREGULAR]	Sets the font style to Regular
[FONTBOLD]	Sets the font style to Bold
[FONTITALIC]	Sets the font style to Italic
[FONTBOLDITALIC]	Sets the font style to Bold Italic
[FONTCOLOR= <i>colorname</i>]	Sets the font color to <i>colorname</i> . 16 valid color names are available: Black, Maroon, Green, Olive, Navy, Purple, Teal, Gray, Silver, Red, Lime, Yellow, Blue, Fuschia, Aqua and White.
[FORECOLOR= <i>colorname</i>]	Sets the Foreground color to <i>colorname</i> .
[BACKCOLOR= <i>colorname</i>]	Sets the Background color to <i>colorname</i> .
[FONTCOLORRGB= <i>r,g,b</i>]	Sets the font color to the specified RGB color values. (Red Green & Blue i.e. <i>r,g</i> & <i>b</i> may range from 0 to 255)
[FORECOLORRGB= <i>r,g,b</i>]	Sets the foreground color to the specified RGB color.
[BACKCOLORRGB= <i>r,g,b</i>]	Sets the background color to the specified RGB color.
[TRANSPARENT]	Sets the background color to transparent.
[UNDERLINE=ON]	Enables text underlining.
[UNDERLINE=OFF]	Disables text underlining.
[STRIKEOUT=ON]	Enables text strikeout.
[STRIKEOUT=OFF]	Disables text strikeout.
[HORIZONTALDPI= <i>n</i>]	Sets the printer Horizontal Dots Per Inch value to <i>n</i> .
[VERTICALDPI= <i>n</i>]	Sets the printer Vertical Dots Per Inch value to <i>n</i> .
[DisplayASCII]	Switches the display character set to ASCII.
[DisplayANSI]	Switches the display character set to ANSI.
[PDFSET=DisplaySet]	Sets the character set used to generate PDF symbols to the same character set used to display text on screen.
[PDFSET=ASCII]	Sets the PDF417 character set to ASCII.
[PDFSET=ANSI]	Sets the PDF417 character set to ANSI.
[PDFMODULEWIDTH= <i>n</i>]	Sets the Module Width for PDF417 to <i>n</i> using the currently selected measurement units.
[PDFMODULEHEIGHT= <i>n</i>]	Sets the Module Height for PDF417 to <i>n</i> using the currentlyselected measurement units.
[PDFMODE=AUTOEBN]	Sets the PDF417 data compression mode to Automatic (EXC, Binary and Numeric modes auto select) i.e. maximum compression.
[PDFMODE=AUTOEB]	Sets the PDF417 data compression mode to Automatic (EXC and Binary modes auto select). Switching to Numeric mode is disabled.
[PDFMODE=EXC]	Sets the PDF417 compression mode to EXC.
[PDFMODE=BIN]	Sets the PDF417 compression mode to Binary.
[PDFMODE=NUM]	Sets the PDF417 compression mode to Numeric.

[PDFCODEWORDS=ON]	Enables display of generated PDF417 code words.
[PDFCODEWORDS=OFF]	Disables display of generated PDF417 code words.
[PDFSECURITY=AUTO]	Sets the PDF417 security level to Auto
[PDFOVERHEAD=<i>n</i>]	Sets the percentage of PDF417 data security overhead when the PDF417 security level is set to Auto. <i>n</i> = the percentage of the symbol area to be used for error correction overhead.
[PDFSECURITY=<i>n</i>]	Sets the PDF417 security level to <i>n</i> . (0-8)
[PDFMAXROWS=<i>n</i>]	Sets the maximum number of data rows for PDF417 symbols to <i>n</i> . The value for <i>n</i> may range from 3-90.
[PDFMAXCOLS=<i>n</i>]	Sets the maximum number of data codeword columns for PDF417 symbols to <i>n</i> . The value <i>n</i> may range from 3-30
[PDFASPECT=<i>n</i>]	Sets the PDF417 symbol aspect ratio to <i>n</i> .
[TRUNCATEDPDF=ON]	Enables generation of truncated PDF417 symbols.
[TRUNCATEDPDF=OFF]	Disables generation of truncated PDF417 symbols.
[LEFTPRINTMARGIN=<i>n</i>]	Sets the left or top margins for where bar codes printed from B-Coder will appear on the printed page. The value <i>n</i> should be supplied in inches. For example the command [TopPrintMargin=1.25] instructs B-Coder to print all bar codes 1.25 inches from the top of a page.
[TOPPRINTMARGIN=<i>n</i>]	
[PRINT]	Prints the currently displayed bar code on the default Windows printer. The Print command will print the bar code at the page position using the current LEFTPRINTMARGIN and TOPPRINTMARGIN settings and it will also eject the page from the printer.
[PRINTNOEJECT]	Prints the currently displayed bar code on the default Windows printer using the current LEFTPRINTMARGIN and TOPPRINTMARGIN settings without eject the page from the printer.
[EJECTPAGE]	Ejects the page from the printer after printing bar codes using the [PRINTNOEJECT] command.
[HORIZONTALDPI=<i>n</i>]	Sets the printer Horizontal Dots Per Inch value to <i>n</i> .
[VERTICALDPI=<i>n</i>]	Sets the printer Vertical Dots Per Inch value to <i>n</i> .
[RASTERIMAGERESOLUTION=<i>n</i>]	Sets the number of pixels per inch to use when saving raster images to disk. The value for <i>n</i> may range from 20 to 1440. The default is 300 pixels per inch.
[BITSPIXEL=<i>n</i>]	Sets the number of Bits Per Pixel to use when saving raster images to disk. Valid values for <i>n</i> are 1, 8 or 24. All other values will be ignored.
[PNGINTERLACE=ON]	Enables interlacing of PNG images that are saved to a disk file.
[PNGINTERLACE=OFF]	Disables interlacing of PNG images that are saved to a disk file.
[BMPCOMPRESSION=ON]	Enables RLE compression of all bitmap images saved to a disk file.
[BMPCOMPRESSION=OFF]	Disables RLE compression of all bitmap images saved to a disk file.
[GIFCOMPRESSION=ON]	Enables LZW compression of all GIF images saved to a disk file.
Note: you must also supply a LZW Unlock Key value using the [LZWUNLOCKKEY= <i>n</i>] command in order to enable LZW compression of GIF files.	
[GIFCOMPRESSION=OFF]	Disables LZW compression of all GIF images saved to a disk file.
[TGACOMPRESSION=ON]	Enables RLE compression of all TGA images saved to a disk file.
[TGACOMPRESSION=OFF]	Disables RLE compression of all TGA images saved to a disk file.
[TIFFCOMPRESSION=OFF]	Disables compression of all TIF images saved to a disk file.
[TIFFCOMPRESSION=LZW]	Enables LZW compression of all TIF images saved to a disk file.
Note: you must also supply a LZW Unlock Key value using the [LZWUNLOCKKEY= <i>n</i>] command in order to enable LZW compression of TIF files.	
[TIFFCOMPRESSION=PACKBITS]	Enables compression of all TIF images saved to a disk file using the PACKBITS compression algorithm.
[TIFFCOMPRESSION=GROUP3]	Enables compression of all TIF images saved to a disk file using the GROUP3 compression algorithm.
[TIFFCOMPRESSION=GROUP4]	Enables compression of all TIF images saved to a disk file using the GROUP4 compression algorithm.

[PNGALLFILTERS]	Enables the use of all compression filters for PNG images saved to a disk file.
[PNGNOFILTER]	Disables the use of compression filters for PNG images saved to a disk file.
[PNGSUBFILTER]	Enables the SUB compression filter for PNG images saved to disk.
[PNGUPFILTER]	Enables the UP compression filter for PNG images saved to a disk.
[PNGAVGFILTER]	Enables the AVERAGE compression filter for PNG images saved to disk.
[PNGPAETHFILTER]	Enables the PAETH compression filter for all PNG images saved to disk.
[JPEGQUALITY=<i>n</i>]	Sets the JPEG image quality for all JPG images that are saved to a disk file. The value for <i>n</i> is expressed as a percentage and may range from 1 to 100.
[LZWUNLOCKKEY=<i>n</i>]	Sets the Unlock Key value used to unlock the LZW compression capabilities in B-Coder for saving GIF and TIF images to a disk file. The value for <i>n</i> must be obtained from TAL Technologies after you purchase a LZW license from Unisys Corporation.

DDE Examples on the Internet

The following pages contain several macro examples for Microsoft Word, and Access that demonstrate how to automate the process of adding bar codes to these applications. Many more examples can also be found in Support section of the TAL Technologies web site located at: <http://www.taltech.com>. The examples on the web site will be the latest up to date versions and may contain many advanced features that are not found in the samples in this manual. The examples on the web site are also saved in the correct file format for each specific application so that you will not have to type them into your computer making them much easier to work with.

We strongly recommend that you visit our web site before starting any development process where you will be calling B-Coder using DDE from another application. The web site also has an extensive technical support section where you will find answers to the most commonly asked technical support questions.

DDE Examples for Microsoft WORD

Note: A Word document template named "B-Coder Word Macros.DOT" will automatically be installed in your B-Coder directory when you run the SETUP program for B-Coder. This template contains all of the following Word macros along with several other useful macros so that you do not have to type them into Word. To install all macros, open the file B-Coder Word Macros.DOT (located in your B-Coder folder) with Microsoft Word and follow the instructions in the template.

Changing Selected Text To A Bar Code:

The following Word macro copies the currently selected text to the clipboard, generates a bar code using the text and pastes the bar code into the document replacing the original text.

Sub TextToBarCode()

' Note: B-Coder must be running for this subroutine to work.

Selection.Copy ' Copy the currently selected text to the clipboard

Chan = DDEInitiate("B-Coder", "System") ' Initiate DDE link with B-Coder

DDEExecute Chan, "[Paste/Build/Copy]" ' Create the bar code using data in the clipboard

DDETerminate Chan ' Terminate the DDE link

Selection.PasteSpecial DataType:=wdPasteMetafilePicture, Placement:=wdInLine

' Paste the bar code into the Word document.

End Sub

Prompting The User For A Bar Code Message

The following macro pops up a dialog box asking for input of a bar code message. After you enter a bar code message, the macro launches B-Coder, sends it the bar code message that you entered, generates the bar code, closes B-Coder and finally pastes the bar code from the clipboard to the open document in the Microsoft Word window.

Sub PromptForBarCode()

```
Msg$ = InputBox("Enter A One Line Bar Code Message", "Bar Code") ' get bar message
If Len(Msg$) Then
  Shell "C:\B-CODER3\B-CODER.EXE", vbMinimizedNoFocus ' Run B-Coder Minimized
  ' Note: make sure that the path for B-Coder is correct in the above command
  Chan = DDEInitiate("B-Coder", "System") ' Initiate link with B-Coder
  DDEExecute Chan, "[PrintWarnings=off]" ' Turn off printer messages
  DDEExecute Chan, "[MessageWarnings=off]" ' Turn off message warnings
  DDEExecute Chan, "[Barcode=" + Msg$ + "]" ' Generate the bar code
  DDEExecute Chan, "[APPEXIT]" ' Tell B-Coder to quit
  DDETerminate Chan ' remove the DDE Link
  Selection.PasteSpecial DataType:=wdPasteMetafilePicture, Placement:=wdInLine
End If
End Sub
```

Note: B-Coder can be forced to load a specific configuration file when it is launched by specifying a configuration file name on the command line used to activate it. For example, you could first create a B-Coder configuration file that sets up B-Coder to use a specific bar code symbology and/or any other parameters. You could then launch B-Coder with the name of your configuration file on the command line so that it would be already configured for your particular application when it is launched. Suppose you wanted to modify the above macro so that it automatically starts up pre-configured to use the PostNET bar code symbology. You could create a configuration file for B-Coder with the PostNET symbology selected and then save the configuration file with the name POSTNET.BCF. To pre-load the POSTNET.BCF configuration file in the above macro, you would simply change the 4th line to read:

```
Shell "C:\B-CODER\B-CODER.EXE POSTNET.BCF"
```

DDE Example for Microsoft Access

The following page contains an example MS Access function that fills up a picture field (an OLE object field) in an MS Access database table with bar codes. The function reads through the entire table and takes the source text for each bar code from a text field named "ItemID" and passes it to B-Coder. Next, it generates the bar code and finally pastes the resulting bar code into the picture field named "Picture" in the same record as each "ItemID" field.

The function below assumes that a table named "Products" is available that has at least two fields, one being a text field named "ItemID" (containing our source bar code text data) and another empty OLE object field named "Picture" (the destination for all bar code graphics).

After you get all your bar codes into your Access table as picture objects, you can print them in reports, display them in forms or use them however you want. This procedure also places the bar codes into the table as Windows Metafiles instead of as bitmaps so you will get a much higher resolution printout and each graphic will use much less space in your database.

Note: The following macro example can also be found in a file named " README.TXT" that will automatically be installed in your B-Coder directory when you run the setup program for B-Coder. This allows you to simply copy the macro from the README.TXT file instead of having to type the macro into Access.

Function BarCodes ()

' dimension all variables

Dim Total As Long, x As Long, Chan As Long, MyBarCodePictureField As String

Dim BCDirectory As String, MyTable As String, MyBarCodeTextField As String

```
BCDirectory = "C:\B-Coder3"           ' Directory where B-Coder lives
MyTable = "Table1"                   ' Name of Access table containing bar code data
MyBarCodeTextField = "ItemID"        ' Field name containing the text for all bar codes
MyBarCodePictureField = "Picture"    ' Name of the OLE object field (Picture field)
                                     ' where bar codes are to be inserted

On Error Resume Next                  ' ignore all errors
Chan = DDEInitiate("B-Coder", "System") ' Establish link.
If Err Then                           ' If error occurs, B-Coder isn't running.
  Err = 0                              ' Reset the error
  x = Shell(BCDirectory + "\B-Coder.exe", 7) ' and start B-Coder.
  If Err Then Exit Function            ' If another error occurs then exit.
  Chan = DDEInitiate("B-Coder", "System") ' else Establish link and continue.
End If

DDEExecute Chan, "[MESSAGEWARNINGS=OFF]" ' Turn off B-Coder warnings
DDEExecute Chan, "[PRINTWARNINGS=OFF]"

DoCmd.OpenTable MyTable, A_NORMAL     ' Select our Access table
Total = DCount(";", MyTable)         ' How many data records in table?
DoCmd.GoToRecord A_Table, MyTable, A_FIRST ' Go to the first record in the table

For x = 1 To Total                   ' Loop through each record
  DoCmd.GoToControl MyBarCodeTextField ' Select message text field
  DoCmd.DoMenuItem A_FORMBAR, A_EDITMENU, A_COPY, , A_MENU_VER20
                                     ' Copy to clipboard
  DDEExecute Chan, "[Paste/Build/Copy]" ' Pass it to B-Coder & build a bar code
  DoCmd.GoToControl MyBarCodePictureField ' Select the bar code picture field
  DoCmd.DoMenuItem A_FORMBAR, A_EDITMENU, A_PASTE, , A_MENU_VER20
                                     ' Paste in the bar code image

  DoCmd.GoToRecord A_Table, MyTable, A_NEXT ' Goto next record
Next                                     ' Loop through all records
DDETerminate Chan                       ' Terminate the DDE link - all done!
End Function
```

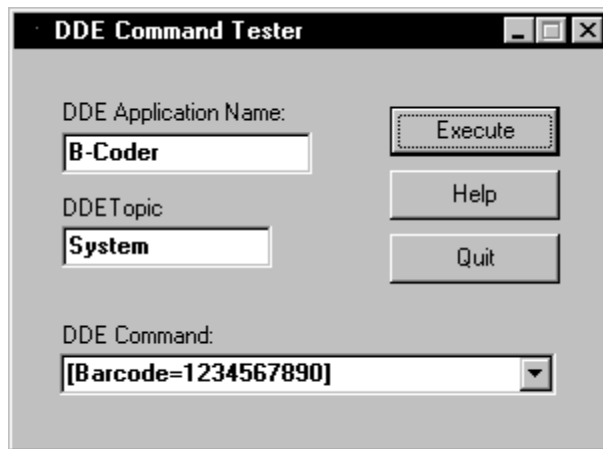
Note: In all versions of Access prior to Access 95, the Access Basic "DoCmd" statement requires a single space immediately after it instead of a period as shown above. For example, use the syntax: **DoCmd OpenTable MyTable\$, A_NORMAL** in place of **DoCmd.OpenTable.MyTable\$, A_NORMAL** when using Access 2.0

Using the LINKTEST Utility to test DDE Commands

The B-Coder comes with a utility called LINKTEST that allows you to test all B-Coder DDE commands (or test commands that can be sent to other programs). B-Coder must be running either in a window or as an icon in order to process DDE commands.

LINKTEST provides two text boxes where you specify the DDE Application Name and a DDE Link Topic for the application that you want to send DDE commands to. The defaults for these parameters are those required by B-Coder.

A third text box labeled "DDE Command" is where you enter the DDE Command that you want to issue to either B-Coder or another program. You can have LINKTEST issue the command to the specified application by clicking your mouse on the button marked "Link Execute".



If you get an error message then you either did not specify the correct parameters for the DDE Application name or Topic or the application is not currently running or else it does not recognize the command you are trying to have it execute.

For additional information about using LINKTEST including a list of all DDE Commands that B-Coder can recognize, click the button marked "Help" in the LINKTEST window.

If you use LINKTEST with an application other than B-Coder, you will have to refer to it's users manual to find out the proper DDE parameters (Application Name and Link Topic) and also the proper syntax of any DDE commands that it can recognize.