B-CODER Professional HELP INDEX

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 dimension of your bar codes. These fields will differ depending on the features of each chosen bar code symbology.

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.

Note: If you press the Enter key while editing a bar code message, B-Coder will insert a carriage return linefeed pair. These characters will appear as a musical note and a rectangle with a circle in it 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.

If the bar code symbology that you have chosen supports characters that cannot be typed in on your keyboard, an ASCII chart will also appear so that you can choose characters that cannot be typed on your PC's keyboard.

To enter data in your bar code text using the ASCII chart, move the text cursor in the bar code text edit field to the point where you want to insert a character and then select it by double clicking your mouse on the character in the ASCII chart. The chosen 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.

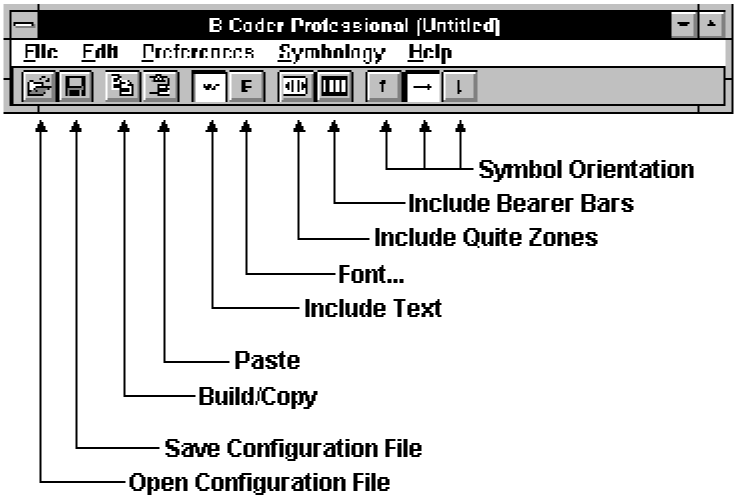
Note: The Code 128 and EAN/UCC 128 symbologies supports not only the full 128 character ASCII character set, but they also supports special *Function Codes.* These function codes can be found at the bottom of the ASCII chart when either of these symbologies is selected

See Also:

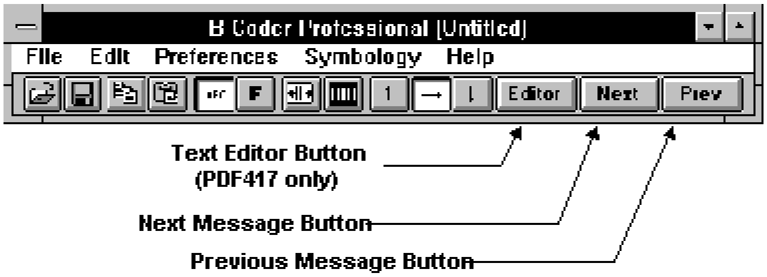
The Main Menu
Bar Code Dimensions
Symbology Descriptions and Rules.

The B-Coder Toolbar

The toolbar in B-Coder's window provides quick access to some of the more useful commands inside 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 is the current symbology, a button marked **Editor** will appear that allows access to a text editor for editing long bar code messages.



Special Considerations and Incompatibilities

B-CODER has been thoroughly tested with a wide variety of Windows 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 most of these problems have simple solutions. Four general types of problems have been identified as listed below.

Pasting Bar Codes Into Programs That Work Only With BitMaps
Pasting Bar Codes Into A SpreadSheet
Printing Bar Codes On A Dot Matrix Printer
Display and Printer Driver Incompatibilities

See Also:

How To Produce Readable Bar Codes

Pasting Bar Codes Into Programs That Only Work With BitMaps

B-CODER produces a Windows MetaFile graphics image for all bar codes. MetaFiles are scaleable images that consist 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. Bit maps on the other hand do not automatically use the highest resolution possible when printed and therefore the printed output of a bit map will retain the same (lower) quality as that of your computer monitor.

When you paste a bar code into a graphics or drawing program that only has the capability of working with bit map files (.BMP) such as the PAINTBRUSH program that comes with Windows, your bar code will be automatically converted from a MetaFile format to a bit map format. During the conversion, the bar code will be deformed and may become unreadable when it is printed.

A possible solution to this problem is to increase the **Narrow Bar Width** value in the B-CODER window and avoid the use of any scaling features in the target application (i.e. SHRINK & GROW in PaintBrush). Acceptable bar codes can be produced with these types of programs but their output should always be tested for readability.

See Also: Bar Code Dimensions

Pasting Bar Codes Into A Spreadsheet

A 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 be solved simply by turning grid lines off before printing the spreadsheet. In Excel this is done by selecting "Options" and "Display..." from Excel's main menu and then clearing the "Grid lines" check box.

Spreadsheets also will normally draw a border around the image both while it is displayed on screen and when it is printed on a printer. The problem with this is that 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 the bar code will be unreadable. The easiest way to avoid this problem is to enable the Include Quiet Zones option in B-CODER's Edit menu.

Another solution would be to remove unwanted borders with some sort of formatting command in the spreadsheet. In Microsoft Excel for example you can get rid of borders around graphics by selecting "Format" & "Patterns" from Excel's main menu. This will display a dialog box where you can instruct Excel to remove a border from a selected graphic.

Printing Bar Codes On A Dot Matrix Printer

Dot matrix printers offer the lowest resolution of all the available printer types. 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 essentially 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 width of a printer dot because there is no way for a dot matrix printer to print a line that is less than the width of a single dot. The larger the value for the Narrow Bar Width, 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 even allow you to set the intensity or darkness of the printing. If your printer driver supports this feature, you should set the darkness level to the darkest intensity that is supported.

Finally, when printing all bar codes on any printer, you should always test the readability of your bar codes with all reading devices that you intend to use.

See Also:

Bar Code Dimensions
How To Produce Readable Bar Codes
The Measurement Units Command

Bar Code Dimensions

Because of differences in 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 symbol 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 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 bar code, whichever is greater. UPC and EAN bar codes have more specific size requirements if they are to be used for product identification.

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.

See Also:

How To Produce Readable Bar Codes
The Measurement Units Command
Bar Code Symbology Descriptions and Rules (UPC/EAN)

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 select 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 element ratio of 2.5.

Note: Higher quality bar code readers may be able to successfully read bar codes with a narrow to wide element 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 successfully. Because of 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.

See Also:

Bar Code Dimensions
Bar Code Symbology Descriptions and Rules.

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 that allows you to specify the magnification factor as a percentage of the standard element width. (80% to 200%)

See Also:

Bar Code Dimensions
How To Produce Readable Bar Codes
Bar Code Symbology Descriptions and Rules
The Measurement Units Command

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.

See Also:

Additional PDF417 Size Control Options
Bar Code Symbology Descriptions and Rules (PDF417)

PDF417 Module Width

The best choice for the PDF417 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.

See Also:

Additional PDF417 Size Control Options
Bar Code Symbology Descriptions and Rules (PDF417)

PDF417 Module Height

The recommended value for the PDF417 Module Height is approximately three times the value for the Module Width however the symbol specification allows for module heights as small as 10 mils (3 printer dots on a 300 DPI laser printer).

See Also:

PDF417 Module Width
PDF417 Aspect Ratio
Bar Code Symbology Descriptions and Rules (PDF417)

PDF417 Aspect Ratio

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 possible 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: Bar Code Symbology Descriptions and Rules (PDF417)

Bar Code Basics

Bar code is an automatic identification technology that allows data to be collected rapidly and with extreme accuracy. Bar codes provide a simple and easy method of encoding information that is easily read by inexpensive electronic devices.

A bar code consists of a series of parallel, adjacent bars and spaces. Predetermined width patterns (symbologies) are used to code 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 into the original data by simple electronic circuits. Due to the design of most bar code symbologies it does not make any difference if you scan a bar code from right to left or from left to right.

There are a variety of different encoding schemes or "symbologies", each of which were originally developed to fulfill a specific need in a specific industry. Several of these symbologies have matured into de-facto standards that are used universally throughout most industries. The symbologies supported by B-CODER are those most commonly used across all industries.

See Also: Bar Code Symbology Descriptions and Rules

How To Produce Readable Bar Codes

Although there are many different types of bar codes, they all share some simple requirements in order to be readable by commercially available bar code scanners.

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 printed on white paper yield the best results. If you intend to use colored ink or colored paper, you should always test the readability of your bar codes before committing to a color scheme. Red ink produces the poorest readability.

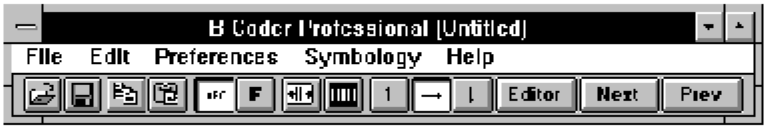
There should always be a small amount of space or *Quiet Zones* preceding and following the bar code so that the reader is able to properly determine the true start and end of the bar code symbol. A good rule of thumb is to reserve at least a quarter of an inch (6.4mm) or 10 times the width of a single narrow bar (whichever is greater) for blank space at either end of a bar code.   B-CODER will automatically include Quiet Zones of 10 times the narrow bar width if the Include Quiet Zones option is enabled in the EDIT 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 entirely unacceptable bar codes. 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.

See Also:

Bar Code Dimensions
Printing to Dot Matrix Printers

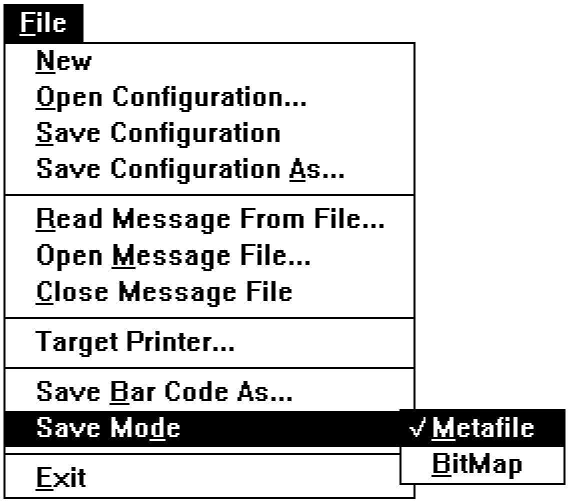
The Main Menu



B-CODER's main menu contains the following sub menus: FILE, EDIT, PREFERENCES, SYMBOLOGY and HELP. For more information about specific sub menu commands select one of the following:

File Menu Commands
Edit Menu Commands
Preferences Menu Commands
Symbology Menu Commands

File Menu Commands



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.

The three commands: **Read Message From File...**, **Open Message File...** and **Close Message File** allow you to open and close Bar Code Message Files.

The command Target Printer allows you to specify the dots per inch resolution of the printer that you will eventually use to print your bar codes on.

The next two commands: **Save Bar Code As...** and **Save Mode** can be used to save bar code graphics to a disk file. See Also: Saving Bar Codes to Disk

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

B-Coder Configuration Files

When you set up B-Coder for a particular application, you can save all the settings that you choose from B-Coder's menus in a "configuration file" so that when you exit B-Coder, you can start up where you left off the next time you run B-Coder.

Four commands are available in the File menu for creating and saving configuration files:

The **New** command causes B-Coder to reset itself to its default configuration.

The **Open Configuration** command allows you to open a previously saved B-Coder configuration. 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 easily enter the filename and directory path where you would like your configuration file to be saved.

Bar Code Message Files

B-Coder can read data from disk files to be used as the source of your bar code messages. The File 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 PDF417 bar code symbology. Because PDF417 can encode up to 2725 characters in a single bar code, a typical PDF417 application might be to read in the contents of a disk file instead of manually typing in all the data for a PDF417 symbol.

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 simply 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 the center of 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 the contents of 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.

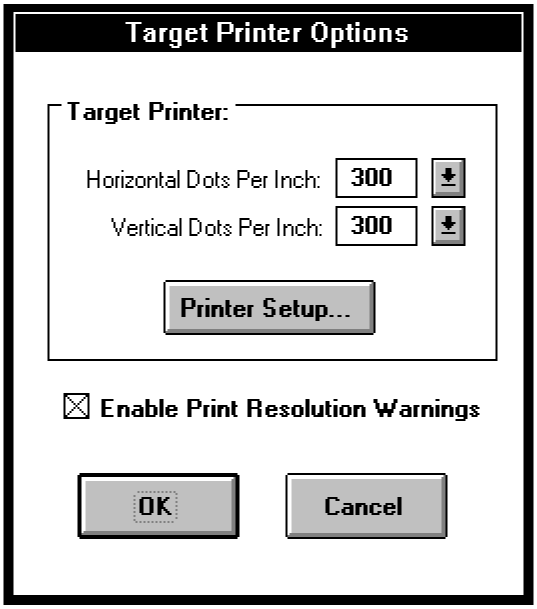
The Target Printer Command

The File menu command **Target Printer** 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 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 that is 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.

Another reason for this feature is because 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 units of printer dots essentially guarantees that you will end up with perfectly readable bar codes no matter what type of printer you will be using.

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



Target Printer Options

Target Printer:

Horizontal Dots Per Inch: 300

Vertical Dots Per Inch: 300

☒ Enable Print Resolution Warnings

The values that appear by default for the horizontal and vertical dots per inch will be those of the currently selected default printer (chosen from the Windows Control Panel Printer Icon).

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 simply 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 **Printer Setup...** option to select the specific printer from a standard Windows printer selection dialog box.

When you select a particular printer using the Printer Setup option, B-Coder will automatically calculate the correct horizontal and vertical dots per inch values for the chosen printer.

The check box labeled **Enable Print Resolution Warnings** allows you to enable or disable B-Coder's printer resolution warning messages.

See Also:

Bar Code Dimensions
Invalid Message and Printer Resolution Warnings
The Measurement Units Command

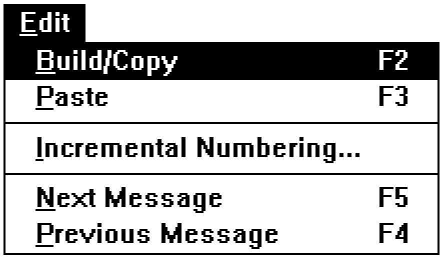
Saving Bar Codes To Disk

The two File menu commands: **Save Bar Code As...** and **Save Mode** can be used to save bar code graphics to a disk file. The command **Save Bar Code As...** opens a standard Windows "File Save As..." dialog box where you can enter a filename for the bar code graphic. The **Save Mode** command allows you to select the graphics mode for the saved bar code.

B-Coder can save bar codes to a file as either Windows **Metafile** graphic images or as **Bitmap** graphic images. Metafiles consist of a sequence of drawing instructions that are used by Windows programs to reproduce a graphic image exactly as the image was originally created. Metafiles are both smaller and much more accurate than any other type of graphic representation. Bitmaps on the other hand consist of an array of pixels that are used to represent an image. The only advantage to saving a bar code as a Bitmap is that Bitmaps are more portable from one operating system to another.

Note: Images often get distorted when they are converted from a Metafile to a Bitmap and thus bar codes saved as Bitmaps may not always be readable. Also, the file size of a Bitmap image is always much larger than an equivalent Metafile. It is highly recommended that you save your bar codes as Metafiles if possible.

Edit Menu Commands



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

The Build/Copy Command

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

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.

See Also: The Paste

The Paste Command

The **Paste** command allows you to paste text from the Windows clipboard into the bar code message text edit control. 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 graphics 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.

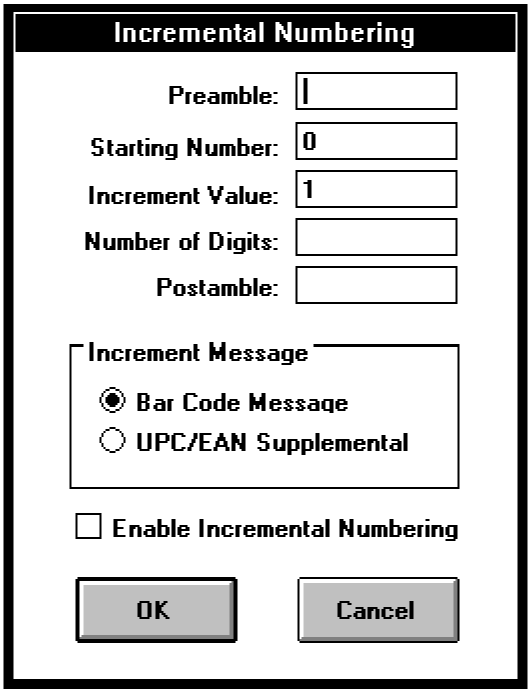
See Also: The Build/Copy Command

The Incremental Numbering Command

The **Incremental Numbering...** command in the Edit menu opens up a dialog box that allows you to 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:



Incremental Numbering

Preamble:

Starting Number: 0

Increment Value: 1

Number of Digits:

Postamble:

Increment Message

● Bar Code Message

○ UPC/EAN Supplemental

□ Enable Incremental Numbering

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

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 contents of a message file.

Incremental Numbering Parameters

When setting up incrementally numbered bar codes messages B-Coder allows you to specify 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** 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.

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.**

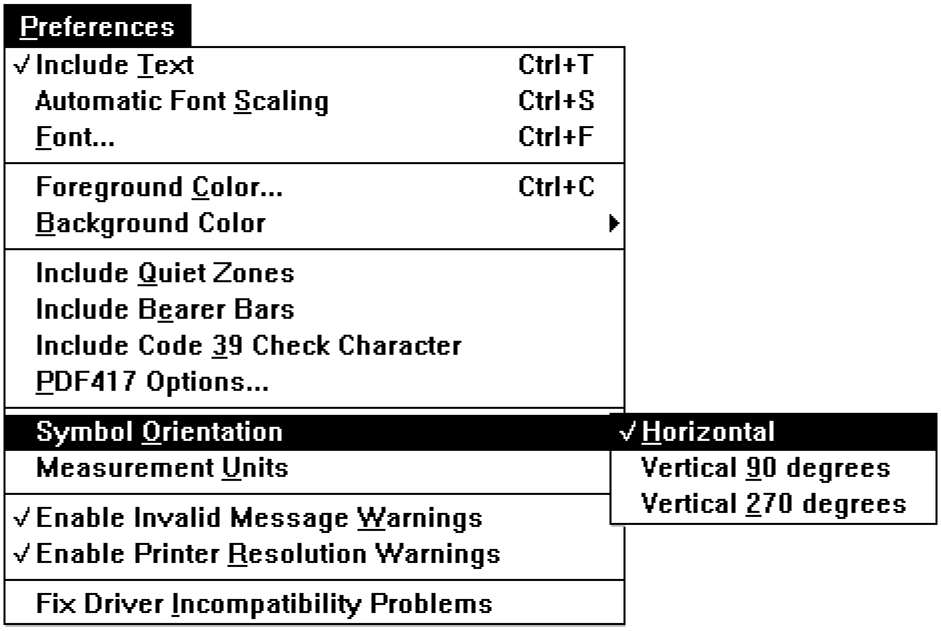
The check box marked **Enable Incremental Numbering** in the **Incremental Numbering** dialog box must be checked to activate incremental numbering.

See Also: The Incremental Numbering Command

The Next Message and Previous Message Commands

When **Incremental Numbering** is active or when a **Bar Code Message File** is open, the Edit menu commands Next Message and Previous Message are used to scroll forward and backward through the sequence of messages. Two buttons labeled **Next** and **Prev** will also appear in the toolbar below B-Coder's main menu. To scroll forward or backward through the messages simply click the **Next** or **Prev** buttons or press function key **F5** to step forward one message or function key **F4** to step backward one message.

Preferences Menu Commands



The **Preferences** menu contains the following choices: Include Text, Automatic Font Scaling, Font..., Foreground Color..., Background Color, Include Quiet Zones, Include Bearer Bars, Include Code 39 Check Character, PDF417 Options..., Symbol Orientation, Measurement Units, Enable Invalid Message Warnings, Enable Printer Resolution Warnings and Fix Driver Incompatibility Problems.

All of the commands in the Preferences menu except the **Font**, **Foreground Color**, **Background Color** and **PDF417 Options...** 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 Command

The Include Text command causes B-Coder to include the human readable version of your bar code message neatly centered along the bottom edge of the bar code graphics image. This option does not apply to PDF417 or Zip+4 bar code symbols.

See Also: The Font... Command

The Automatic Font Scaling Command

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

See Also:

The Font Command
The Include Text Command

The Font Command

The Font option causes B-Coder to display a standard Windows font selection dialog box that allows you to select the type font to be used for all human readable text printed along with the bar code symbol. The font selection dialog box also allows you to select a font style, size, color and strikeout or underline effects for the chosen type font.

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

Note: Different fonts behave differently and some fonts may even appear differently on screen than when they are printed. True Type fonts are generally the most WYSIWYG under Windows and they also seem to align better when rotated 90 or 270 degrees.

Note: When the Automatic Font Scaling option is in effect, it will override any font size selection made from the font selection dialog box.

See Also: The Include Text Command

The Foreground/Background Color Commands

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 that will allow you to 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: B-Coder will allow you to specify any color you want for both the foreground and background of a bar code symbol. Because of this, it is entirely possible to choose color combinations that render a bar code symbol completely 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 solid background colors with a luminescence value no less than 180. The only other advice that we can give is to **always test colored bar codes** before committing to a specific color scheme.

See Also: How To Produce Readable Bar Codes

The Include Quiet Zones command

The Include Quiet Zones command causes B-Coder to include a small amount 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 correctly be able to determine the true beginning and end of a bar code symbol.

Note: Most bar code symbology specifications require quiet zones, therefore it is highly recommended that you enable this option.

See Also: How To Produce Readable Bar Codes

The Include Bearer Bars command

The **Include Bearer Bars** command causes B-Coder to draw lines that surround a bar code symbol.

The purpose of the bearer bar is to equalize the pressure exerted by the printing plate over the entire surface of the symbol and also to enhance the reading reliability by reducing of 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 either 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 Code 39 Check Character command

The Include Code 39 Check Character command causes B-Coder to automatically calculate and append a check character to all code 39 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 bar codes. This feature only applies to Code 39 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.

See Also: Symbology Descriptions and Rules (Code 39)

The PDF417 Options... Command

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.

See Also:

The PDF417 Options Dialog Box
Symbology Descriptions and Rules (PDF417).

The Symbol Orientation Command

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 Command

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 but for the sake of completeness 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.

Invalid Message and Printer Resolution Warnings

The **Enable Invalid Message Warnings** and **Enable Printer Resolution Warnings** options in B-Coder's Preferences menu causes 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
Target Printer Selection
Symbology Descriptions and Rules

The Fix Driver Incompatibility Problems Command

The Fix Driver Incompatibility Problems option   is used to correct display and printer incompatibility problems. B-Coder will work correctly with all video display drivers that come with Windows however some third party video drivers and some printer drivers are unable to display or print bar codes generated by B-Coder correctly.

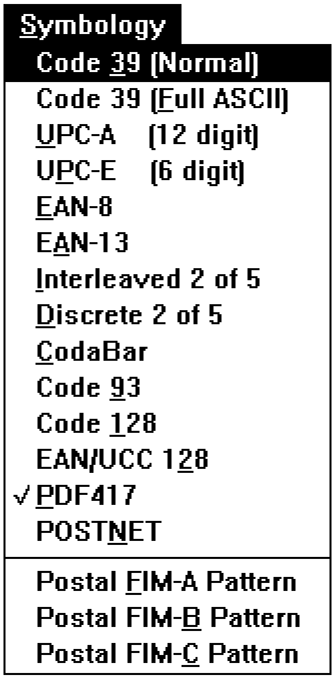
If a complete bar code does not appear when you select BUILD/COPY from B-Coder's Edit menu, try correcting the problem by enabling this option. If this problem occurs on your system, then the problem is with your video driver - not with B-Coder.

The best way to deal with this problem is to use the video drivers supplied with Windows. To change video drivers, run the *Windows Setup* program in the *Main* program group of the Windows Program Manager and select one of the video drivers that were originally supplied with Windows.

Some printer drivers have also been found to produce peculiar looking bar code symbols due to bugs in the printer driver. What typically happens when using one of these "buggy" drivers is that the bars in a symbol are printed as hollow boxes. Again, selecting the Fix Driver Incompatibility Problems option will usually cure this type of dilemma.

See Also: Special Considerations and Incompatibilities

Symbology Menu Commands



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

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 Symbology Descriptions and Rules
Bar Code Dimensions

Getting Started

B-Coder is an extremely easy to use program that generates professional quality bar codes of 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 this program may be scaled to any size and can be used by any Windows program that supports pasting of Windows MetaFile graphics images. (i.e. Microsoft Word, Windows Write, Excel, Lotus 123, Word Perfect, Micrografx Draw, Pagemaker, Corel Draw, Visio, etc.) B-Coder gives you the ability to quickly incorporate bar code technology 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 even supports a set of DDE commands that allow other Windows programs to completely control the operation of B-Coder. 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 a powerful bar code labeling system by writing one simple WordBASIC macro. See Also: DDE Example For MS Word

B-CODER supports all common bar code symbologies including UPC-A, UPC-E, CodaBar, Code-39 Normal, Code 39 Full ASCII, EAN-8, EAN-13, Interleaved-2 of 5, Discrete 2 of 5, Code 93, Code 128, EAN/UCC 128, PDF417 and the U.S. Postal Service POSTNET & FIM. For a complete description of each of the different symbologies, see the section of this document Bar Code Symbology Descriptions and Rules.

To use this program you simply specify the message or number that you want represented as a bar code and select Build/Copy from the Edit menu. Your bar code will be generated and displayed and then automatically copied into the Windows clipboard so that you can paste it directly into any other Windows program. It's that easy!

See Also:

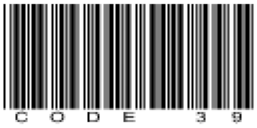
Using B-Coder With Other Windows Programs
The Main Menu
Edit Menu Commands
Bar Code Basics
Understanding Dynamic Data Exchange
DDE Commands Supported By B-Coder

Bar Code Symbology Descriptions and Rules

B-Coder Supports the following bar code symbologies:

CODE 39 (Normal Version)
CODE 39 (Full ASCII Version)
UPC-A, UPC-E and UPC Supplimentals
EAN-8, EAN-13 and EAN Supplimentals
INTERLEAVED 2 of 5
Discrete 2 of 5
CODABAR
CODE 93
CODE 128
EAN/UCC 128
PDF417
POSTNET
Postal FIM Patterns

CODE 39 (Normal version)



The NORMAL CODE 39 is a variable length symbology that can encode the following 44 characters: 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ-. *$/+%.

Code 39 is the most popular symbology in the non retail world and is used extensively in manufacturing, military, and health 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 CODE 39 version and your bar code text contains lower case characters, B-Coder will convert them to upper case. If your bar code message contains any invalid characters, B-Coder will prompt you with a warning message.

Code 39 optionally allows for a 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 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.

See Also: Code 39 Full ASCII Version

Code 39 (Full ASCII Version)

The FULL ASCII version of Code 39 is a modification of the NORMAL (standard) Code 39 symbology 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 supported 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	Code39	ASCII	Code39	ASCII	Code39	ASCII	Code39
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

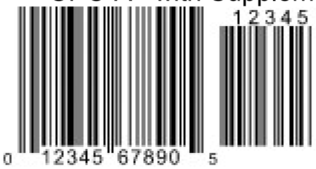
Note: Because all of the characters used to implement Full ASCII Code 39 are part of the Normal Code 39 character set, readers that do not support Full ASCII Code 39 will still read Full ASCII Code 39 symbols. The reader will output shifted characters as if they were normal Code 39 characters. For example 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, a normal Code 39 reader blindly outputs **%H**.

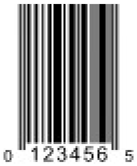
See Also: Code 39 Normal Version

UPC-A, UPC-E & UPC Supplimental

UPC-A with Supplemental

UPC-E





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 UCC at 8163 Old Yankee Road, Suite J, Dayton, OH 45458 Tel: 513-435-3870

When specifying a UPC-A bar codes, you may enter up to 11 digits. B-Coder will calculate the check digit. If you enter less than 11 digits, B-Coder will pad zeros to the left of the number so that the total length is 11. If you enter a message that contains digits other than 0 to 9, B-Coder will warn you with an error message. If you enter more than 11 digits, B-Coder will truncate the number to 11 digits

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.

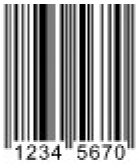
For UPC-E bar codes, you may enter up to 6 digits. Again, B-Coder will calculate the check digit. If you enter a number greater or less than 6 digits wide, B-Coder will truncate or pad the number to a total length of 6 digits.

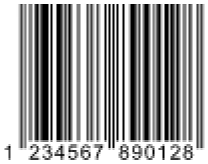
Both UPC-A and UPC-E allow for a supplemental two or five digit number to be appended to the main bar code symbol. This supplemental message was designed for use on publications and periodicals. If you enter a supplemental message, it must consist of either two or five numeric digits. The supplemental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

See Also: UPC/EAN Magnification Factors

EAN-8, EAN-13 & EAN Supplimental

EAN-8 EAN-13





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

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. The check digit is automatically calculated by B-Coder. If you enter non numeric digits, B-Coder will display a warning message. If you enter less than 7 digits, B-Coder will Left-Pad the number with zeros to a width of 7 digits. If you enter a number with more than 7 digits, B-Coder will truncate the number to 7 digits.

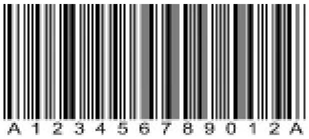
EAN-13 is the European version of the UPC-A symbol. The difference between EAN-13 and UPC-A is that EAN-13 encodes a thirteenth digit into the parity pattern of the left six digits of a normal UPC-A symbol. This thirteenth digit, in combination with the twelfth digit, defines two flag characters that represent a country code. For compatibility with UPC, flags 00, 01, 03, 04, and 06 through 09 are assigned to the U.S.

Both EAN-8 and EAN-13 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 bar code numbers are assigned to specific products and manufacturers by an organization called **ICOF** located in Brussels, Belgium. Tel: 011-32-2218-7674

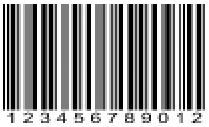
See Also: UPC/EAN Magnification Factors

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 CodaBar characters and starts and ends with a valid start/stop character. If you specify lower case letters for A B C or D, B-Coder will convert to upper case.

INTERLEAVED 2 OF 5

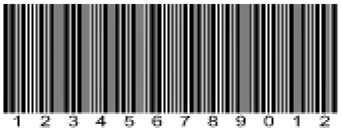


Interleaved 2 of 5 is a very high density variable length numeric only symbology that encodes digits in an interleaved manner. The odd position digits are encoded in the bars and the even position digits are encoded in the spaces.

Because of this, Interleaved 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 probability 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-determined 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 padded with leading 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.

Discrete 2 OF 5



Discrete 2 of 5 is 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 dense as Interleaved 2 of 5 and also, odd numbers of digits may be encoded. Use of Discrete 2 of 5 is not very common therefore few bar code readers support this symbology.

See Also: Interleaved 2 of 5

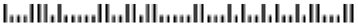
CODE 93

CODE 93 is a variable length symbology that encodes 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 as Code 39. 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 include them as part of your bar code message.

POSTNET



POSTNET (**POST**al **N**umeric **E**ncoding **T**echnique) is a 5, 9 or 11 digit numeric only bar code symbology used by the U.S. 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 control over the size of POSTNET bar codes.

NOTE: Most other Windows programs allow you to modify the size of metafile graphics 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.

See Also: Postal FIM Patterns

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. One of three different start characters tell the bar code reader which of the character sets is initially being employed and three shift codes permit changing character sets inside a printed symbol.

One Code 128 character sets encodes all upper case characters and all ASCII control characters, another encodes all upper and lower case characters and the third set encodes numeric digit pairs 00 thru 99. This third character set effectively doubles the code density when printing numeric data. Code 128 also employs a check digit for data security.

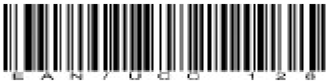
In addition to ASCII characters, Code 128 also allows encoding of four function codes (FNC1 - FNC4). The meaning of function code FNC1 and FNC4 were originally left open so that bar code application designers could use these extra codes for their own application specific purposes. Recently an agreement 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 open 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 bar code symbol, any data also contained in the symbol is discarded.

Code 128 function codes can be found at the bottom of the ASCII chart in B-Coder's main window when Code 128 is the currently selected symbology.

B-Coder will automatically select the proper character sets and insert the proper character set 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.

See Also: EAN/UCC 128

EAN/UCC 128



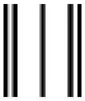
The EAN/UCC 128 symbology is a variation of the original Code 128 symbology designed primarily for use in product identification applications. The EAN/UCC 128 specification uses the same code set as Code 128 except that it does not allow function codes FNC2-FNC4 to be used in a symbol and FNC1 is used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

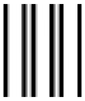
Code 128 function codes can be found at the bottom of the ASCII chart in B-Coder's main window when Code 128 is the currently selected symbology.

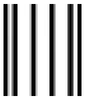
See Also: Code 128

Postal FIM Patterns

Fim A FIM B FIM C







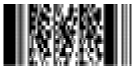
FIM or Facing Identification Mark patterns are another type of postal bar code used in automated mail processing by the U.S. Postal service. FIM patterns are used for automatic facing and cancelling of mail that does not contain a stamp or meter imprint (business reply mail, penalty mail, etc.). They also provide a means of seperating 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 mail that is not preprinted with Postnet bar codes. FIM-C is used on business reply, penalty and franked mail (government 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 on 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.

See Also: POSTNET

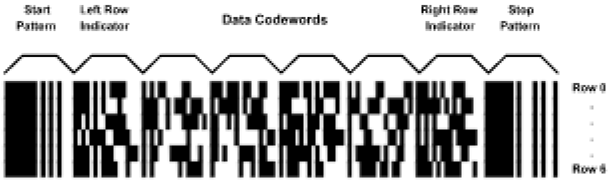
PDF417



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 compression options, error detection and correction options, and variable size and aspect ratio symbols. The symbology was published by Symbol Technologies 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 compression 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 compression algorithms may be used, it is possible for different printed symbols to be created from same input data.

The symbology also allows for varying degrees of data security or error correction and detection. Nine different security levels are available with each higher level adding additional overhead to the printed symbol.

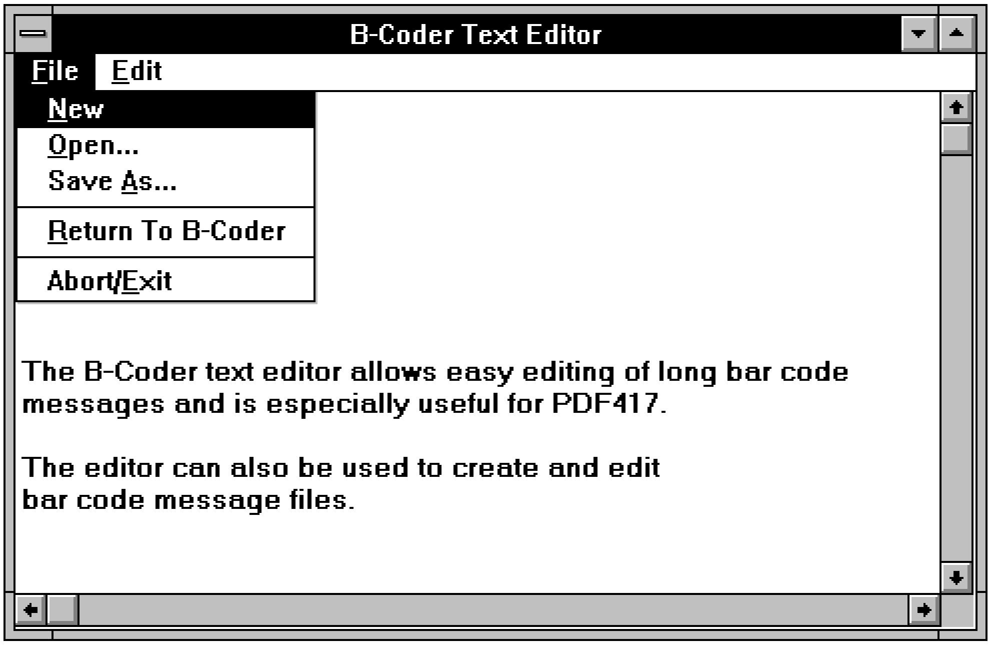
B-Coder allows complete control over all optional features of PDF417 by selecting PDF417 Options... from the Preferences menu.

See Also:

PDF417 Bar Code Dimensions
The PDF417 Options Dialog Box
PDF417 Data Compression Modes
PDF417 Error Detection And Correction
Truncated PDF417 Symbols
B-Coder's PDF417 Text Editor

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 automatically 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 main window. Shown below is the B-Coder PDF417 text editor.



The B-Coder text editor allows easy editing of long bar code messages and is especially useful for PDF417.

The editor can also be used to create and edit bar code message files.

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 standard edit commands **Cut**, **Copy** and **Paste**.

Note: The text editor is designed for editing text messages only. B-Coder will not allow you to edit binary files or any data containing ASCII control codes other than carriage returns, linefeeds or tabs.

See Also: Symbology Descriptions and Rules (PDF417)

PDF417 Data Compression Modes

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

1. **Extended Alphanumeric Compaction** mode (**EXC**) This mode supports encoding of all printable ASCII characters and can compress approximately 2 characters per code word.

2. **Binary/ASCII Plus** mode. This 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 thru 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 code words to switch between modes within the symbol. B-Coder gives you the option to select any of the three standard data compression 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 compression modes and provides the maximum data compression possible. The second *auto* 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.

See Also:

The PDF417 Options Dialog Box
Symbology Descriptions and Rules (PDF417)

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 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 or *security* levels numbered 0 thru 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 thus reducing the maximum number of data characters that can be encoded in a symbol and also increasing the size of a printed symbol. (A single PDF417 symbol can contain a maximum of 928 total code words for both data and error correction.)

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

Security Level	**Erasures + (2 * Misdecodes)**	**Additional Overhead 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

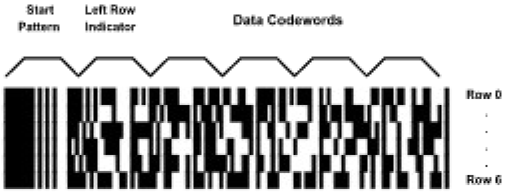
See Also:

The PDF417 Options Dialog Box
Symbology Descriptions and Rules (PDF417)

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:

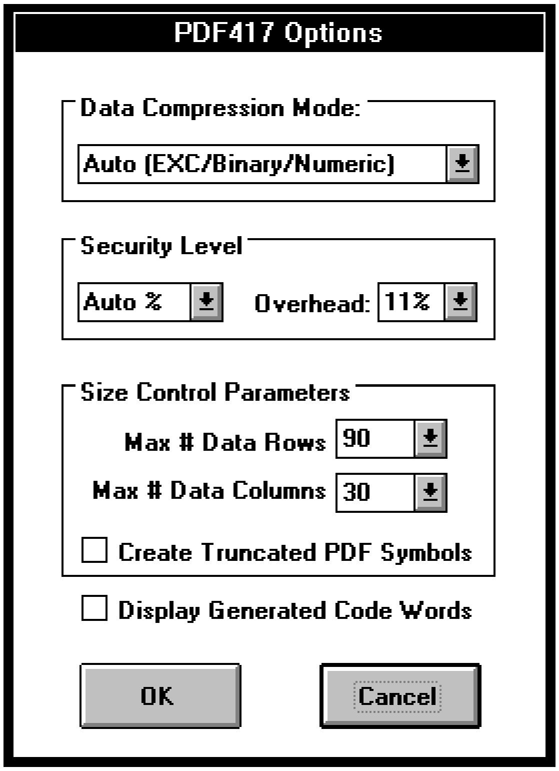


See Also:

The PDF417 Options Dialog Box
Symbology Descriptions and Rules (PDF417)

The B-Coder PDF417 Options Dialog Box.

B-Coder allows for a great deal of control over how a PDF417 symbol is 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:



PDF417 Options

Data Compression Mode:
Auto (EXC/Binary/Numeric)

Security Level
Auto %   Overhead: 11%

Size Control Parameters
Max # Data Rows 90
Max # Data Columns 30
□ Create Truncated PDF Symbols

□ Display Generated Code Words

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

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 PFD417 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.

See Also:

Symbology Descriptions and Rules (PDF417)
PDF417 Bar Code Dimensions

PDF417 Data Compression Mode Options

The B-Coder PDF417 Options Dialog Box allows you to select a data compression mode for all PDF417 symbols. 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.

See Also: PDF417 Data Compression Modes

PDF417 Security Level Options

The B-Coder PDF417 Options Dialog Box allows you to select a security level from 0 to 8 for all PDF417 symbols.

You can also have B-Coder automatically select a security level based on a percentage of total symbol area to be used for error correction. If you select **Auto%** for the Security Level, a list box will appear where you can choose the percentage of error correction overhead to add from 0% to 99%.

The Auto% Security 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 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%.

See Also:

Symbology Descriptions and Rules (PDF417)

Additional PDF417 Size Control Options

The The PDF417 Options Dialog Box allows you to select additional size control options for all PDF417 symbols. 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:

10mils x [(17 x MaxCols) + 69]<=2000 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 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 per row.

See Also:

PDF417 Bar Code Dimensions
Symbology Descriptions and Rules (PDF417)

Using B-Coder With Other Windows Programs

To use B-CODER with other Windows programs simply 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 Bar Code Text edit control 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 other program's Edit menu. The bar code should then appear in the program's window.

Do not be alarmed if the bar code does not look exactly right on your computer's screen. Your monitor's resolution may or may not be high enough to completely render the bar code exactly as it will appear when you finally print the bar code on your printer.

See Also:

The B-CODER Window
The Main Menu

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 application. 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 if the server was designed to accept them. The types of commands that a server program can accept will vary depending on the server. B-Coder understands 107 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 commands. For example, you could write a program in a word processor's macro language that fully automates the process of 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 link is always initiated by the client application. A client can usually initiate a DDE link using some form of **DDEInitiate** command in it's macro or programming language. The *Initiate* command is essentially a querry containing a **DDE Application Name**, A **DDE Topic**, and optionally a **DDE Item** to all other applications currently running. The Application Name, Topic, and Item 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 Application Name and 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.

See Also:

DDE Commands Supported By B-Coder
DDE Example For Microsoft Word

DDE Commands Supported By B-Coder

B-Coder supports a set of 106 different DDE commands that can be issued by other Windows programs to completely take control over it. 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.

See Also:
Understanding Dynamic Data Exchange
DDE Example For Microsoft Word

[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.

[QUIETZONES=ON]
Enables automatic Quiet Zones around all bar codes.

[QUIETZONES=OFF]
Disables automatic Quiet Zones around all bar codes.

[CODE39CHECK=ON]
Enables Code 39 Check character for Code 39 bar codes.

[CODE39CHECK=OFF]
Disables Code 39 Check character for Code 39 bar codes.

[TEXT=ON]
Enables human readable text below bar codes.

[TEXT=OFF]
Disables human readable text below bar codes.

[BUILD/COPY]
Generates a bar code and copies it to clipboard.

[PASTE]
Pastes text from the clipboard into the bar code message

[CODE39]
Selects Code 39 as default symbology.

[CODE39FULLASCII]
Selects Code 39 Full ASCII as default symbology.

[UPCA]

Selects UPC-A as default symbology.

[UPCE]
Selects UPC-E 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.

[CODABAR]
Selects CodaBar as default symbology.

[CODE93]
Selects Code 93 as default symbology.

[CODE128]
Selects Code 128 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.

[MESSAGE='text',n,...]
Changes the bar code message to the supplied *text* 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 ***Test*** followed by a carriage return (Ascii 13), and a line feed (Ascii 10).

[SUPPLEMENT='text']
Changes UPC/EAN supplemental to the supplied *text*.
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 ***12345***.

[BARWIDTH=n]
Changes the value for the narrow bar width to *n*.

[BARHEIGHT=n]
Changes the value for the bar code height to *n*.

[MAGNIFY=n]
Changes the UPC/EAN magnification factor to *n*

[RATIO=n]
Changes the Narrow To Wide Element Ratio to *n*

[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.

[PDFMODULEWIDTH=n]
Sets the Module Width for PDF417 to *n* using the currently selected measurement units.

[PDFMODULEHEIGHT=n]
Sets the Module Height for PDF417 to *n* using the currently selected measurement units.

[PDFMODE=AUTOEBN]
Sets the PDF417 data compression mode to Automatic (EXC, Binary and Numeric modes auto select) i.e. maximum ompression.

[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=n]
Sets the percentage of PDF417 data security overhead when the PDF417 security level is set to Auto.
n= the percentage of the symbol area to be used for error correction overhead.

[PDFSECURITY=n]
Sets the PDF417 security level to *n*. (0-8)

[PDFMAXROWS=n]
Sets the maximum number of data rows for PDF417 symbols to *n*. (3-90)

[PDFMAXCOLS=n]
Sets the maximum number of data codeword columns for PDF417 symbols to *n*. (3-30)

[PDFASPECT=n]
Sets the PDF417 symbol aspect ratio to *n*.

[TRUNCATEDPDF=ON]
Enables generation of truncated PDF417 symbols.

[TRUNCATEDPDF=OFF]
Disables generation of truncated PDF417 symbols.

[MESSAGEWARNINGS=ON]
Enables invalid message warnings.

[MESSAGEWARNINGS=OFF]
Disables invalid message warnings.

[PRINTWARNINGS=ON]
Enables printer resolution warnings.

[PRINTWARNINGS=OFF]
Disables printer resolution warnings.

[INCPREAMBLE=text]
Sets the Message Preamble to *text* for Incrementally Numbered bar code messages.

[INCPOSTAMBLE=text]
Sets the Message Postamble to *text* for Incrementally Numbered bar code messages.

[INCSTART=n]
Sets the starting number to *n* for Incrementally Numbered bar code messages.

[INCVALUE=n]
Sets the Increment Value to *n* 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]
Causes B-Coder to apply incremental numbering to the bar code message.

[INCSUPPLEMENT]
Causes B-Coder to apply incremental numbering to the UPC/EAN supplemental message.

[FIXDRIVER=ON]
Enables fixing of display or printer driver incompatabilities.

[FIXDRIVER=OFF]
Disables fixing of display or printer driver incompatabilities.

[HORIZONTAL]

Sets the symbol orientation to horizontal.

[VERTDOWN]
Sets the symbol orientation to Vertical rotated 90 degrees i.e. reading from top to bottom.

[VERTUP]
Sets the symbol orientation to Vertical rotated 270 degrees i.e. reading from bottom to top.

[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=font]
Sets the fontname for humab readable text to *font*.

[FONTSIZE=n]
Sets the font size to *n* 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=colorname]
Sets the font color to *colorname*. 16 valid color names are available: Black, Maroon, Green, Olive, Navy, Purple, Teal, Gray, Silver, Red, Lime, Yellow, Blue, Fuschia, Aqua and White.

[FORECOLOR=colorname]
Sets the Foreground color to *colorname*.

[BACKCOLOR=colorname]
Sets the Background color to *colorname*.

[FONTCOLORRGB=r,g,b]
Sets the font color to the specified RGB color values. (Red Green & Blue i.e. r,g & b may range from 0 to 255)

[FORECOLORRGB=r,g,b]
Sets the foreground color to the specified RGB color.

[BACKCOLORRGB=r,g,b]
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=n]
Sets the printer Horizontal Dots Per Inch value to *n*.

[VERTICALDPI=n]
Sets the printer Vertical Dots Per Inch value to *n*.

[SAVEMODEMETA]
Causes B-Coder to save bar codes as Metafiles

[SAVEMODEBMP]
Causes B-Coder to save bar codes as BitMaps

[SAVEBARCODE(filename)]
Saves the current bar code symbol to disk with the specified*filename*.

[SAVEBARCODE]
Saves the current bar code symbol to disk with the most recently specified filename.

[NEW]
Resets B-Coder to its default settings.

[OPENCONFIG(filename)]
Opens a B-Coder configuration file with the specified *filename*.

[SAVECONFIG]
Saves the current configuration to disk.

[SAVECONFIG(filename)]
Saves the current configuration to disk with the specified *filename*.

[READMSGFILE(filename)]
Reads in the contents of the specified file as a bar code message.

[OPENMSGFILE(filename)]
Opens the specified file containing multiple bar code messages.

[CLOSEMSGFILE]
Closes any open bar code message file.

DDE Example for Microsoft WORD

The following is a simple example of a Microsoft Word macro that opens a DDE link to B-Coder, changes the bar code message to "Testing", generates the bar code and finally copies the bar code from the clipboard to the open document in the Microsoft Word window.

```
Sub MAIN
Chan = DDEInitiate("B-Coder", "System")    ' Initiate DDE link with B-Coder
DDEExecute Chan, "[Message='Testing']"     ' Change message to "Testing"
DDEExecute Chan, "[Build/Copy]"            ' Generate the bar code
DDETerminate Chan                          ' Terminate the DDE link
EditPaste                                  ' Paste bar code into document
End Sub
```

See Also:

Understanding Dynamic Data Exchange
DDE Commands Supported By B-Coder

Other T.A.L. Software Products

T.A.L. Enterprises has been providing innovative solutions to the automatic identification industry since 1985. In addition to B-Coder Professional for Windows, our growing line of products includes:

B-Coder & B-Coder Source Code For DOS
The Software Wedge For DOS & Windows
Comshow RS232 Diagnostic Software

For more information about any T.A.L. product or to place an order, please call:

T.A.L. Enterprises
2022 Wallace Street
Philadelphia, PA 19130-322222

Tel: **800-SCAN-004** (800-722-6004) or **215-763-2620**   Fax: **215-763-9711**

B-Coder & B-Coder Source Code For DOS

The DOS version of B-CODER uses command line arguments to print exceptionally high quality, scaleable bar codes of various symbologies on Epson or compatible dot matrix printers. The source code for B-CODER for DOS is available.   (Written in Microsoft QuickBASIC)

The Software Wedge For DOS & Windows

Our best selling product is a program called The SOFTWARE WEDGE. The DOS version of this program is a ram resident (TSR) utility whose basic function is to capture data from a serial port on your PC and transfer the data to the DOS keyboard buffer while other foreground applications are running.

The SOFTWARE WEDGE allows any PC program to receive data from a serial (RS232) device as if it were typed in directly on the PC's keyboard. For example, in order to read data from a bar code scanner directly into a spreadsheet or database program you would normally have to purchase a hardware device called a "keyboard wedge" that connects between your PC and it's keyboard. The bar code scanner would then be connected to the "wedge" so that whenever you read a bar code, the data would appear to your computer as if it were coming from the keyboard.

The problem with hardware wedges is that they are expensive, extremely limited in capability, and they are not altogether reliable. What's more, they cannot be used on a laptop or portable PC that does not have a detachable keyboard.

The SOFTWARE WEDGE solves all these problems by allowing you to connect a bar code reader (or any other RS232 device) directly to the serial port on your PC and perform the same function more accurately and for a lot less money.

The SOFTWARE WEDGE offers dozens of advanced features that allow it to work with just about any serial device including electronic scales, measuring instruments, laboratory equipment, temperature sensors, etc.. It even supports bi-directional communications, timed automatic prompting, input data parsing and filtering, automatic date and time stamping, keystroke macro insertion and more. The entire program uses only 5K of RAM and is extremely easy to set up and use.

The SOFTWARE WEDGE FOR WINDOWS performs essentially the same function as the DOS version but is designed for use with any Windows application program. In addition to all the powerful features of the DOS version, the Windows version can also act as a DDE server to other Windows programs that support Dynamic Data Exchange. A New Professional version for Windows is also available that is designed for use with more complicated serial devices like PDF417 bar code scanners.

Comshow RS232 Diagnostic Software

COMSHOW is a RS232 (serial) diagnostics program that is designed as an aid in debugging serial communications problems on a PC. COMSHOW allows you to take complete control of your serial ports down to the hardware level so that you can easily track down any problems that you may be experiencing with RS232 communications between your PC and any RS232 device. COMSHOW is extremely easy to use and is an indispensable tool for anyone who is experiencing problems with their serial ports. This program also comes complete with several extra utility programs that make RS232 system development remarkably easy. COMSHOW is included free of charge with the The Software Wedge For DOS

Choosing A Bar Code Reader

There are essentially three main types of bar code readers currently available: Bar Code Wands or Pen type readers, CCD type bar code readers, and Bar Code Laser Scanners. Each of the three types of readers have their own specific advantages and disadvantages.

Bar Code Wands

Bar Code Wands are the smallest and least expensive of the three main reader types typically ranging in price between $200 to $400. Wands are pen shaped devices that use an LED light source mounted at the tip of the "pen".

To read a bar code, the wand must be physically dragged across the bar code in a smooth steady motion. Because the tip of the wand is usually in contact with the bar code symbol, there is a good chance that the bar code image will get scratched or damaged if it is read with a wand repeatedly. Another disadvantage with wands is that it can sometimes be difficult to read bar codes printed on irregular shaped objects like cans or bottles.

Wands are a good choice when all your bar codes are printed on flat surfaces and the number of times that each bar code is to be read is low and when size and price are major considerations. Wands are typically used in the Video Rental and Air Parcel business.

See Also:

CCD type bar code readers
Bar Code Laser Scanners

CCD Readers

CCD readers are the next step above bar code wands in both price and performance. CCD readers use an array of sensors approximately 2 to 4 inches wide to read a bar code. Because of this, CCD readers are limited to reading bar codes that have an overall width that is less than the width of CCD sensor array. Typical prices for CCD readers are between $400 and $800. They are non contact devices meaning that they do not damage a bar code during the reading process but they must be held within a few inches of a bar code in order to read successfully. CCD readers have better success than wands when reading bar codes printed on irregular shaped surfaces.

CCD readers are a good choice when reading small to medium sized bar codes that might be printed on irregular surfaces. CCD readers are typically used in retail Point of Sale applications.

See Also:

Bar Code Wands
Bar Code Laser Scanners

Laser Scanners

Laser scanners are the most expensive types of bar code reading devices because they use a moving laser beam to perform a read. Like CCD readers, laser scanners are non contact devices but unlike CCD readers, laser scanners are capable of reading a much wider range of bar code sizes over a much wider range of distances.

Laser scanners come in two main varieties, hand held and fixed mount. Hand held laser scanners look similar to CCD readers but are usually slightly smaller and narrower. Most hand helds use a trigger to activate the scanning mechanism which is essentially a reciprocating mirror that rapidly scans the laser beam across the surface of a bar code. Hand held laser scanners must be physically aimed at a bar code so that the moving laser beam scans across the entire bar code perpendicular to the bars and spaces.

Fixed mount laser scanners come in a wide range of styles with the most common being the kind used in grocery store check out lines. Most fixed mount scanners use a sophisticated scanning system that rapidly scans the laser beam in many directions so that the orientation of a bar code with respect to the laser beam does not matter. This allows for extremely rapid reading of many different sized bar codes printed on different shaped objects. Hand held laser scanners are typically priced between $800 and $2000 and fixed mount scanners start at about $800 and go up from there. Laser scanners are excellent for all bar code applications but they really excel when speed or distance is a major factor and when very long bar codes must be read.

See Also:

Bar Code Wands
CCD type bar code readers