# ELFRING FONTS, INC.
# BAR CODE 128

This package includes 24 versions of bar code 128 fonts in scalable TrueType & PostScript formats, plus a Windows utility, Bar128, that helps you make bar codes. There are two versions of the bar code 128 font in this set: standard and a human readable. Each of these types comes in two different versions *(A/B or C)* and six different aspect ratios. The different aspect ratios let you print bar codes with the same height, but at different horizontal character densities.

Bar code 128 was introduced in 1981 and was designed for high-density alphanumeric work. Bar code 128 produces variable length codes and includes three different symbol sets: two that represent the ASCII character set, and a third set of paired digits, which increase the print density of numeric data by a factor of two. These separate symbol sets are referred to as "A", "B", and "C". Bar code 128 requires a checksum. Special variations of bar code 128 include UCC / EAN 128, SCC-14, and SSCC-18.

**Symbol set A** is used to print upper case letters, numbers, and the standard ASCII control characters. **Symbol set B** is used to print upper and lower case letters, numbers, and punctuation. **Symbol set C** encodes pairs of numbers 00 through 99. The **UCC/EAN option** adds an FNC1 character after the start codes for subsets A, B, and C to uniquely identify that bar code as a shipping code. See Tables 2 - 4 in this documentation for a complete description.

SCC-14, the Shipping Container Code, is used on fixed content shipping containers and uses subset C fonts. SSCC-18, the Serial Shipping Container Code is used as an identifier on shipments to track items and uses subset C fonts. Finally, **AutoSense**, switches between subsets B and C to minimize bar code length and also uses subset C fonts.

All bar codes must start with a specific Start character corresponding to the bar code character set (Start A, Start B, or Start C), and must end with a Stop code. You may switch between bar code subsets in the middle of a bar code. Bar code 128 requires a checksum character, placed before the Stop character. Bar code 128 is essentially a fixed width bar code, with the exception of the Stop character.

## INSTALLATION

Please use our BarCD128.exe program to install this package and font set in Windows. The fonts will be installed into your Windows font folder automatically. Our bar code utility program, Bar128.exe, documentation, and matching files will also be installed. The bar code fonts will appear in all Windows font menus. You can select any one of the bar code just as you normally select any other font in your application. For help in building bar codes, run the Bar128.exe utility program. To access the utility program or the documentation, click on Start, Programs, Bar Code 128, and select the program or documentation item you want to view.

## USER ACCESSIBLE FILES

In Windows Vista and Windows 7, plus all future versions of Windows, users are not allowed to access any files stored in the Programs folder. The Visual Basic macros, label templates, and sample Excel spread sheet are placed into a folder named "Bar128" in each user's Documents

area. You can find these files as follows:

Windows XP:  My Documents\Bar128

Vista:            Libraries\Documents\ Bar128

Windows 7:    Libraries\Documents\ Bar128

Windows 8:    Libraries\Documents\ Bar128

## BAR CODE FONTS

This package contains 2 different versions of the bar code 128 font, one for subsets A & B, and a second for subset C. There are standard and human readable versions of each of those bar codes. Finally, each version has six separate variations, to let you control both bar code height and width *(or the aspect ratio)* separate of each other. So this bar code set contains a total of 24 TrueType fonts. They include:

### Chart 1    Subsets A, B, & C

| Typeface Name | Code 128AB a | Code 128AB b | Code 128AB c | Code 128AB d | Code 128AB e | Code 128AB f |
|---|---|---|---|---|---|---|
| Code 128 Subsets A & B | | | | | | |
| Typeface Name | Code 128ABhr a | Code 128ABhr b | Code 128ABhr c | Code 128ABhr d | Code 128ABhr e | Code 128ABhr f |
| Code 128 Subsets A & B Human Readable | A 2 B e | A 2 B e | A 2 B e | A 2 B e | A 2 B e | A 2 B e |
| Typeface Name | Code 128C a | Code 128C b | Code 128C c | Code 128C d | Code 128C e | Code 128C f |
| Code 128 Subset C | | | | | | |
| Typeface Name | Code 128Chr a | Code 128Chr b | Code 128Chr c | Code 128Chr d | Code 128Chr e | Code 128Chr f |
| Code 128 Subset C Human Readable | 12345644 | 12345644 | 12345644 | 12345644 | 12345644 | 12345644 |

Note that bar code 128 needs more characters *(106)* than are available in the standard ASCII character set *(characters 32 through 126 = 94)*. Since we are using a font to produce bar code 128, there must be some method of making the additional characters available. We map some of these characters to the high ASCII *(greater than 128)* positions.
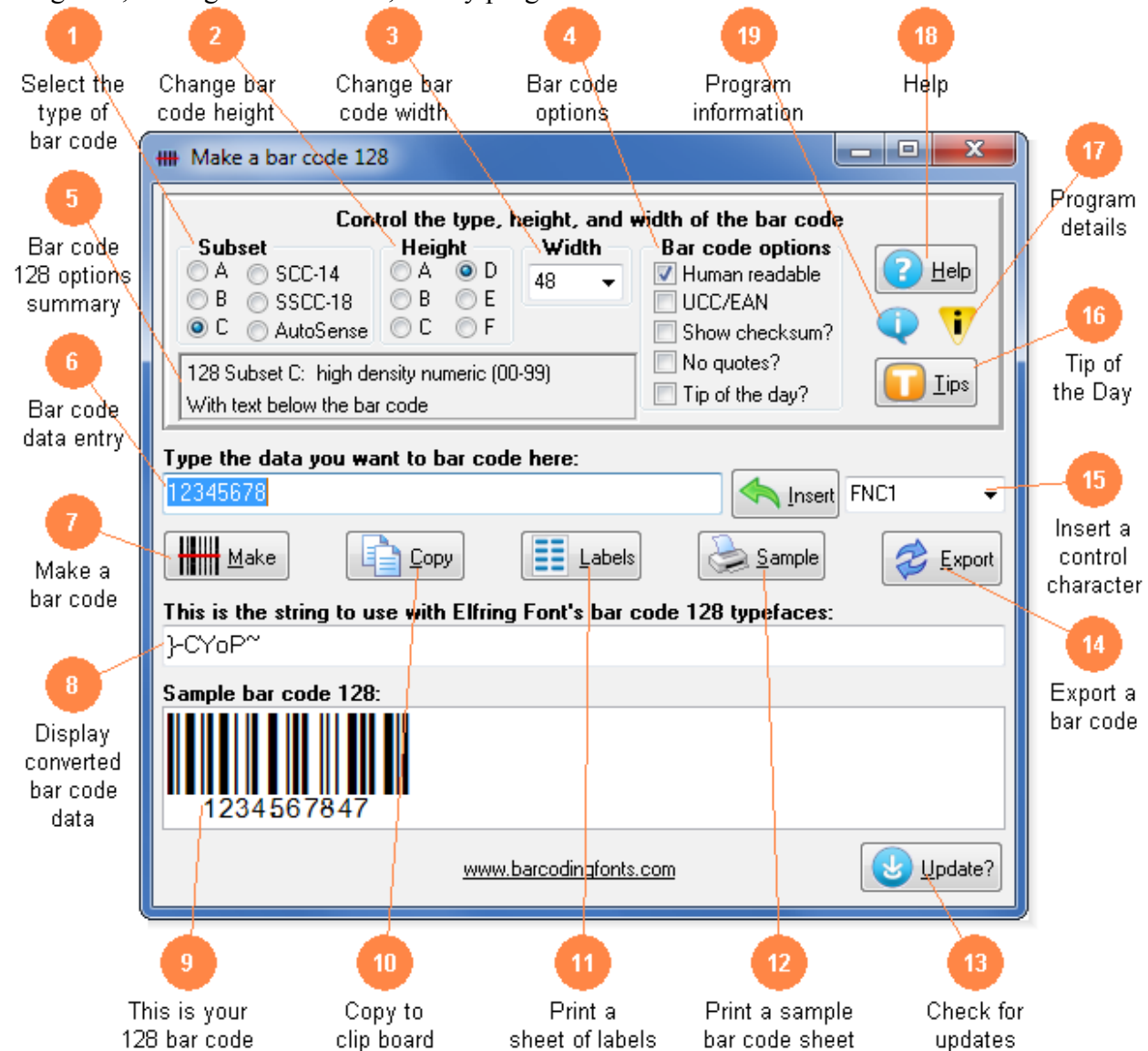
In addition, TrueType fonts do not allow printable character data in the space character. Unfortunately bar code 128 uses the space character, so the space character in these bar code

fonts has been moved to another character location. Finally, because of different encoding restraints there are separate bar code sets for the A/B and the C subsets.

Character Tables 2 - 4, shown on the last three pages of this manual, must be used to work with bar code 128. The mapping of these characters has been chosen to make using these bar codes as simple as is possible, however some compromises had to be made. Each table entry shows the ASCII character you must use to generate a particular bar pattern, the 128 A / B / C code you get, and the bar code 128 Value assigned to this pattern. Bar code 128 Values are especially important since they are used in calculating the checksum.

## THE BAR128 UTILITY PROGRAM

The Bar128 utility program was automatically added to your system when you ran our install program. You can access the Bar128 utility from the Desktop icon, or by clicking on: Start, Programs, Elfring Bar Code 128, Utility program.

**1** Select the type of bar code

**2** Change bar code height

**3** Change bar code width

**4** Bar code options

**19** Program information

**18** Help

**17** Program details

**5** Bar code 128 options summary

**16** Tip of the Day

**6** Bar code data entry

**15** Insert a control character

**7** Make a bar code

**14** Export a bar code

**8** Display converted bar code data

**9** This is your 128 bar code

**10** Copy to clip board

**11** Print a sheet of labels

**12** Print a sample bar code sheet

**13** Check for updates

### Make a bar code 128

**Control the type, height, and width of the bar code**

Subset: A, SCC-14, B, SSCC-18, C, AutoSense
Height: A, D, B, E, C, F
Width: 48
Bar code options: ☑ Human readable, ☐ UCC/EAN, ☐ Show checksum?, ☐ No quotes?, ☐ Tip of the day?

128 Subset C: high density numeric (00-99)
With text below the bar code

Type the data you want to bar code here:
12345678

Insert FNC1

Make  Copy  Labels  Sample  Export

This is the string to use with Elfring Font's bar code 128 typefaces:
}-CYoP~

Sample bar code 128:
1234567847

www.barcodingfonts.com

Update?

3

This utility converts your bar code 128 data into actual bar codes. Select the subset and options you need including SCC-14 or SSCC-18. Then click *Make* to add the appropriate start and stop codes and to calculate your checksum. Use this utility to build bar codes and then to copy and paste them into other Windows programs for printing. Click on any orange circle below to view a description of exactly what that control does.Bar codes can be printed on sheets of labels, copied into a desktop publishing program to display a bar code on your packaging, or exported as a gif, jpg, or png file. These bar code fonts are compatible with virtually any Windows program. For programs like Excel and Access see the package documentation for details on using our Visual Basic macros directly inside of those programs.

1) **Select the subset.**  Select the subset of bar code 128 to use. **Subset A** allows the use of upper case letters, numbers, control characters, and symbols. **Subset B** allows the use of upper and lower case letters, numbers, and symbols. **Subset C** is a high density numeric only encoding method. You must have an even number of digits when using Subset C. **SCC-14** produces Shipping Container Codes and **SSCC-18** produces Serial Shipping Container Codes. **AutoSense** switches between subsets B and C in minimize bar code length.

2) **Change the bar code height.** Modify the height of a bar code without changing the width.

3) **Change the bar code width.** Scale the bar code width up or down by increasing or decreasing the point size.

4) **Bar code options.** Basic bar code 128 options. Switch between human readable and standard bar codes. Human readable bar codes print the data in the bar code below the bar code patterns. Turn UCC/EAN 128 bar coding on or off. EAN 128 bar codes add an FNC1 character after the Start code and is typically used in subset C. Show Checksum displays the values used to calculate a checksum. No Quotes disables the use of the ASCII quote character. Tip of the Day turns this feature on or off at program start up.

5) **Bar code 128 options summary.** This window provides a summary of all of the bar code 128 options you have selected.

6) **Bar code data entry.** Type in the data you want to convert to a bar code here. This program will add the selected Start code to the beginning of your bar code, calculate the checksum for you and append it after your data, and then add the Stop code at the end of your bar code.

7) **Make a bar code.** Click this button to make a bar code from your data.

8) **Display converted bar code data.** This is the string of characters to use with our bar code fonts. Note you can not just type in the bar code 128 fonts. Your bar code data is converted into a data string and this data string must be used with our bar code 128 fonts. Also note that subset C uses a different set of fonts from subsets A and B.

9) **This is your 128 bar code.** This is a sample of what your bar code will look like. Note that Windows screen resolution is fairly low, so the bar code displayed on screen may not exactly match what will be printed.

10) **Copy to clip board.** Copy your completed bar code to the Windows clip board. *(The Windows clip board is invisible- don't worry about where it is or how it works.)*  You can then paste this bar code directly into virtually any Windows program by pressing Ctrl-V or using the Edit, Paste menu in your program.

11) **Print a Sheet of Labels.** Print a sheet of labels in your word processor. Select a label size and our program will open the matching label template in your word processor. Then just paste your bar code on to any label.

12) **Print a sample bar code sheet.** Print a sample page of your bar code to see just what it will look like. This sample page shows your bar code in a number of different sizes.

13) **Check for updates.** Check the web for a program update. You must have an active internet connection running. A web page will be displayed with information about the current version of this program and whether or not you need to update.

14) **Export a bar code.** Export a copy of your bar code in bmp, gif, jpg, or png formats. You can choose from four different resolutions. You can email this bar code or use it in your packaging.

15) **Insert a control character.** All bar code 128 subsets allow the entry of some kinds of characters that can not be typed from a Windows keyboard. To enter one of these special characters, select the character you want to enter from the drop down box on the right. *(Like FNC1)*  Then click the *Insert*  key to enter that character into your bar code.

16) **Show a *Tip of the Day*.** Display the next program *Tip of the Day*. From this display you can page forwards or backwards through the list of all available tips.

17) **Program details.** Display this help screen.

18) **Help.** Launch the Windows Help application. The Help form will give you detailed answers to most questions about the program.

19) **Program information.** Display program and contact information including the version number of this software.

## SCC-14



(01) 0 07 12345 12345 9

| (01) | 0 | 0 | 712345 | 12345 | 9 |
|---|---|---|---|---|---|
| Application Identifier<br><br>Do not enter In our utility Or macros | Packaging Type<br><br>0 = box/carton 1 = pallet 2 = larger | Use a leading 0 for UPC codes or<br><br>1st digit of EAN code | First 6 digits Of UPC code | 5 digit item identification number from UPC | Mod 10 Check digit<br><br>We calculate, Do not add |

The Shipping Container Code, (SCC-14) is used on fixed content shipping containers. Note that this bar code only contains numbers. It has 14 variable digits, and an additional first two that must always be 01. The last digit is a checksum.  The human readable portion shown above is formatted for people to read. The actual bar code does not include and spaces or parenthesis. This bar code uses the **Code 128C** fonts. Assume your UPC code is **712345123459**. To create an SCC-14 bar code from your UPC code:

1) Start with the packaging type, enter a 0 for a box (or a 1 or a 2 for other carton sizes)

2) Enter a second 0, followed by the first digit of your UPC code
3) Enter the next 5 digits of your UPC code
4) Enter the next 5 digits of your UPC code- leaving out the last digit of your UPC
5) Our program will calculate the checksum of 9
6) So your data would be:  0 + 0 + 7 + 12345 + 12345 + 9 = **00712345123459**

## SSCC-18



(00) 00712345 123456789 7

| (00) | 0 | 0 | 712345 | 123456789 | 7 |
|------|---|---|--------|-----------|---|
| Application Identifier | Extension digit<br><br>Set to 0 | Use a leading 0 plus 1$^{st}$ digit of UPC code<br><br>or<br><br>first digits of EAN code | 6 digit UPC company ID number<br><br>or<br><br>digits 2 – 7 of EAN code | 9 digit serial number<br><br>not to be used again for 12 months | Mod 10 Check digit |

The Serial Shipping Container Code, (SSCC-18) bar code is used as an identifier on shipments to track items. Note that this bar code only contains numbers. It has 18 digits, but the first two must always be 00 and the last digit is a checksum. The human readable portion shown above is formatted for people to read. The actual bar code does not include and spaces or parenthesis. This bar code uses the **Code 128C** fonts. Assume your UPC code is **712345678904**. To create an SSCC-18 bar code from your UPC code:

1) Start with a leading 0 extension digit
2) Enter a second 0, followed by the first digit of your UPC code
3) Enter the next 5 digits of your UPC code, leaving out the last 6 digits of your UPC
4) Enter a 9 digit serial number
5) Our program will calculate the checksum of 7
6) Your data would be:  0 + 0 + 7 + 12345 + 123456789 + 7 = **007123451234567897**

### CHECKSUMS

Bar code 128 requires a checksum. A checksum is a special character that is added to your bar code. The checksum helps the bar code reader verify that the bar code is correct. Note that while the checksum character is read by the bar code scanner, it is not passed along as part of your data. The checksum must be printed after your data, and before the Stop code. The checksum is based on a weighted modulo 103 calculation. While this may seem complicated, it is easy to do on a computer. *(See our Bar128 utility program.)*  If you want to use Access, Excel, or Word to print bar codes, see the Visual Basic function documentation starting on page 6. Note that Tables 2 - 4 assign a value from 0 to 105 to each possible bar code 128 character. The checksum is calculated as follows:

1) Initialize a sum variable to the value of your Start *code (Start A = 103, Start B = 104, and Start C = 105).*
2) Initialize the Weighting value to 1.
3) Starting with the first character in your bar code after the Start code *(working from left to right),* look up the V*alue* associated with that character and multiply that value by the Weighting value.
4) Increment the Weighting value by 1, and add the result of the calculation above to your sum variable.

5) Repeat this until there is no more data, then divide the sum variable by 103. The remainder from this calculation is the checksum. Convert the Value to a character via Tables 2 - 4.
6) Put the checksum character after your data and end the bar code with the Stop code character (~).
7) Note that both the sum and the checksum variables for any bar code can be displayed by our Bar128 utility program.

The following examples show checksum calculations for both an A and a C 128 bar code.

## Sample Checksum Calculations

 - Bar code 128 A for **CODE 128**

{CODEä128t~  - ASCII to produce bar code

| Char | Value | Multiplier | Sum |
|---|---|---|---|
| { | Start A | n/a | 103 |
| C | 35 | 1 | 103 + 1*35 |
| O | 47 | 2 | 138 + 2*47 |
| D | 36 | 3 | 232 + 3*36 |
| E | 37 | 4 | 340 + 4*37 |
| Space (ä) | 00 | 5 | 488 + 5*0 |
| 1 | 17 | 6 | 488 + 6*17 |
| 2 | 18 | 7 | 590 + 7*18 |
| 8 | 24 | 8 | 716 + 8*24 |
| t | checksum | | 908/103 = 8 remainder 84 (84 = "t") |
| ~ | Stop | | |

 - Bar code 128 C for **12345678**

}-CYoP~   - ASCII to produce bar code

| Char | Value | Multiplier | Sum |
|---|---|---|---|
| } | Start C | n/a | 105 |
| - | 12 | 1 | 105 + 1*12 |
| C | 34 | 2 | 117 + 2*34 |
| Y | 56 | 3 | 185 + 3*56 |
| o | 78 | 4 | 353 + 4*78 |
| P | checksum | | 665/103 = 6 remainder 47 (47 = "P") |
| ~ | Stop | | |

 - UCC / EAN 128 C for **12234456**

}²-8MYD~      - ASCII to produce bar code

| Char | Value | Multiplier | Sum |
|---|---|---|---|
| } | Start C | n/a | 105 |
| 2 | FNC1 | 1 | 105 + 1*102 |
| - | 12 | 2 | 207 + 2*12 |

| | | | |
|---|---|---|---|
| 8 | 23 | 3 | 231 + 3*23 |
| M | 44 | 4 | 300 + 4*44 |
| Y | 56 | 5 | 476 + 5*56 |
| D | checksum | | 756/103 = 7 remainder 35 (35 = "D") |
| ~ | Stop | | |

 - Subset switching starting in subset C, then switch to subset B for **12345**

}-CÌ6W~   - ASCII to produce bar code

| Char | Value | Multiplier | Sum |
|---|---|---|---|
| } | Start C | n/a | 105 |
| - | 12 | 1 | 105 + 1*12 |
| C | 34 | 2 | 117 + 2*34 |
| Ì (switch B) | 102 | 3 | 185 + 3*100 |
| 6 (5, but in C font this is a 6) | 21 | 4 | 485 + 4*21 |
| W | checksum | | 569/103 = 5 remainder 54 (54 = "W") |
| ~ | Stop | | |

 - Subset switching starting in subset B, then switch to subset C for **bc1234**

|cdË-CÊ~   - ASCII to produce bar code

| Char | Value | Multiplier | Sum |
|---|---|---|---|
| | | Start B | n/a | 104 |
| c (b, but in C font this is a c) | 66 | 1 | 104 + 1*66 |
| d (c, but in C font this is a d) | 67 | 2 | 170 + 2*67 |
| Ë (switch C) | 99 | 3 | 304 + 3*99 |
| - | 12 | 4 | 601 + 4*12 |
| C | 34 | 5 | 649+ 5*34 |
| Ê | checksum | | 819/103 = 5 remainder 98 (98 = "Ê") |
| ~ | Stop | | |

## ASPECT RATIOS / PRINT DENSITY

This bar code 128 font set includes six different aspect ratios of each bar code: .25 *(A),* .5 *(B),* .75 *(C),* 1.0 *(D),* 1.5 *(E),* and 2.0 *(F).*  When you are building a bar code, start with the D version of the bar code font. Once you have the bar code length set to what you want, you can vary the bar code height, without changing the bar code length, by changing the font to one of the A, B, C, E, or F versions. The B version will print a bar code of exactly the same length as the D, but at .5 times the bar height. The E version will print a bar code of exactly the same length as the D, but at 1.5 times the bar height. Maximum print density *(not including Start/Stop codes)* is 6.8 characters/inch on 300 dpi printers, or 10.2 CPI on 600 dpi printers. If you exceed this print

density, your bar code scanner may not be able to read the bar codes you print.

## PRINTING SHEETS OF LABELS

This bar code font set adds the ability to print bar codes to other, existing Windows programs. It does not print bar code labels all by itself. There are two easy ways to print bar code labels using our bar code fonts.

You can use Word's address label templates to print sheets of the same bar code. This is a simple way to print an entire sheet of identical bar code labels with no other text. We include a step by step set of instructions, in pdf format, for doing this. Click on *Start, Programs, Bar Code 128, How to Print Labels*  to launch the file in your pdf viewer, or open the file printlabels.pdf with your pdf viewer (the file is located in the User Data Folder).

The second way to print label sheets requires a label template. A template tells your word processor how to arrange information to fit specific types of label sheets. Once you have a template you can use any word processor to build your own labels with bar codes. The bar codes do not have to be identical and you can also include other text or graphics on each label. This package includes templates in RTF format *(compatible with virtually all word processors)* for most popular label styles. These templates are located in your User Data Folder. You can use the **Print Labels** button in our Bar128 utility to create a bar code and select one of these bar code templates, or you can open one of these templates in your word processor to lay out a label. To locate these files using your word processor, click on Open and then select the User Data Folder. Look for a subfolder with the name: \Bar128.

| Template | Label Size | Labels/Sheet |
|---|---|---|
| ef167.rtf | 1.75 x 0.50 | 80 |
| ef570.rtf | 1.75 x 1.25 | 32 |
| ef060.rtf | 2.625 x 1.00 | 30 |
| ef161.rtf | 4.00 x 1.00 | 20 |
| ef162.rtf | 4.00 x 1.33 | 14 |
| ef163.rtf | 4.00 x 2.00 | 10 |
| ef197.rtf | 4.00 x 1.50 | 12 |
| ef164.rtf | 4.00 x 3.33 | 6 |

Our Bar128 utility also lets you create your own custom label template. See your word processor manual for details on how to create a template, or download one from your label supplier. Once you have your own custom template, save the template file *(in rich text format)* in the User Data Bar128 folder with the file name "custom.rtf". Our Bar128 utility can directly access that template to let you easily print labels.

## EMBEDDING BAR CODE FONTS IN PDF FILES

The fonts in this bar code set are **not** embeddable. When you embed a font in a PDF document, you are actually distributing a copy of that font with every single PDF file you generate. Your PDF generator glues a copy of the bar code font to the end of each PDF file it builds. This font is then installed on every computer that views the PDF document.

Embeddable versions of these fonts are available, at additional cost. Pricing for this add-on font set is based on how many computers the embeddable fonts will be installed on and how many

people will view the PDF files. When your purchase an embeddable add-on font set you receive a new version of these fonts, and the new fonts will embed in PDF files. You must contact Elfring Fonts to order an embeddable font set.

## Visual Basic Macros for Access, Excel, and Word

This package contains Visual Basic macros that let you automatically build bar code 128 strings in Excel, Access, and Word. The macro source code and sample Excel 2003 and 2010 spread sheets are located in your user data folder.  This set includes the following functions:

### Table 1

| Function | Details |
|---|---|
| AutoSense(Text) | Converts the input text data into a complete bar code 128, using subset s B & C to minimize bar code length. The function adds the Start code, appends the data, calculates and adds the checksum, and puts the Stop code at the end. This result must be formatted with one of the Code 128C typefaces. |
| Bar128A(Text) | Converts the input text data into a complete bar code 128, subset A. The function adds the Start code, appends the data, calculates and adds the checksum, and puts the Stop code at the end. This result must be formatted with one of the Code 128AB typefaces. |
| Bar128B(Text) | Converts the input text data into a complete bar code 128, subset B. The function adds the Start code, appends the data, calculates and adds the checksum, and puts the Stop code at the end. This result must be formatted with one of the Code 128AB typefaces |
| Bar128C(Text) | Converts the input text data into a complete bar code 128, subset C. The function adds the Start code, throws away all non-numeric data, adds a leading zero if there aren't an even number of digits in the data, converts the numeric data into number pairs, calculates and adds the checksum, and puts the Stop code at the end. This result must be formatted with one of the Code 128C typefaces |
| Bar128Aucc(Text) | Converts the input text data into a complete UCC/EAN bar code 128, subset A. The function adds the Start code, the FNC1, appends the data, calculates and adds the checksum, and puts the Stop code at the end. This result must be formatted with one of the Code 128AB typefaces |
| Bar128Bucc(Text) | Converts the input text data into a complete UCC/EAN bar code 128, subset B. The function adds the Start code, the FNC1, appends the data, calculates and adds the checksum, and puts the Stop code at the end. This result must be formatted with one of the Code 128AB typefaces |
| Bar128Cucc(Text) | Converts the input text data into a complete UCC/EAN bar code 128, subset C. The function adds the Start code, the FNC1, throws away all non-numeric data, adds a leading zero if there aren't an even number of digits in the data, converts the numeric data into number pairs, calculates and adds Code 128C typefaces |
| SCC14(Text) | Converts the input text data into a complete SCC-14 bar code. This function adds the Start code, the FNC1, the AI of 01, throws away all non-numeric data, converts the numeric data into number pairs, calculates and adds the mod 10 and mod 103 checksums and then the Stop character. Your data must be exactly 11 digits long. |
| SCC14checkdigit(Text) | Returns just the mod 10 SCC-14 check digit |
| SSCC18(Text) | Converts the input text data into a complete SSCC-18 bar code. This function adds the Start code, the FNC1, the AI of 00, throws away all non-numeric data, converts the numeric data into number pairs, calculates and adds the mod 10 and mod 103 checksums and then the Stop character. Your data must be exactly 15 digits long. |
| SSCC18checkdigit(Text) | Returns just the mod 10 SSCC-18 check digit |

### Warning!

Unlike our utility program, *Bar128*, these Visual Basic functions do **very** limited error checking. You must make sure that the data you send to the function is correct! If you send bad data, your bar codes may be unreadable or they may not encode the data you think they have. Please use the *Bar128* utility program to verify that your data is correct before using these Visual Basic functions to mass produce bar codes.

### Mail Merge Note

When mail merge data is fed from an Excel spread sheet *(or probably an Access data base)* to

Word, Word loses data that contains quote characters! To work around this problem, we have duplicated the bar code pattern for the quote character and placed it in an additional character position- â. Our Visual Basic macros automatically use this special character, â, in place of the quote character to avoid this problem.

## Using Visual Basic Functions in Excel

Open the spread sheet you want to add bar code 128 functions to *(or create a new spread sheet)*. Click on Tools, Macros, then Visual Basic Editor. In the Visual Basic Editor tool, click on File, Import File, and select the drive and folder where you installed our Bar Code 128 package *(probably Bar128)*. The Visual Basic file **VBbar128.bas** should appear there. Select this file and open it. This will add a new module, EFBAR128, to your spread sheet. This module adds the six functions *(see Table 1)* to your spread sheet and is saved along with it.

These bar code functions can be used in any formula or cell to build working bar codes. For example, if cell H9 is defined as a text cell *(Format, Cell, Number, Text)* and cell I9 has the formula, =Bar128B(H9), then any text entered in cell H9 will be converted into a bar code string in cell I9. Note that you also need to select the proper typeface for that bar code type, using Format, Cell, Font. See Chart 1 for applicable font names.

## Using Visual Basic functions in Access

Open the database you want to add bar code 128 functions to *(or create a new database)*. Under your database Objects, click on Modules, then click on the New icon at the top of the box. This will bring up the Visual Basic Editor tool. Click on File, Import File, and select the drive and folder where you installed our Bar Code 128 package *(probably Bar128)*. The Visual Basic file **VBbar128.bas** should appear there. Select this file and open it. This will add a new module, EFBAR128, to your database. This module adds the six functions to your database and is saved along with it. Table 1 lists these functions.

These bar code functions can be used in a report to build working bar codes. To use these functions, pass data to them from your table fields and return the result in a report. You do this by entering a formula =Bar128A([table.field]) in the control source field of the report. Note that you need to select the proper typeface for that bar code subset. See Chart 1 for a complete listing of font names.

## Using Visual Basic functions in Word

Word does not have the ability to directly use Visual Basic functions. You can, however, combine Word and Excel to print bar code 128! The basic idea is to keep your data in Excel, and to have Excel format the data for bar code printing. You then do a mail merge in Word, selecting the formatted string from Excel as the data source.

Place your data in Excel in a specific column. You then apply one of our bar code functions to the data in that column, producing a second column that contains the formatted string. This second column is passed to Word in the mail merge. You must make sure to select the proper subset typeface for your mail merge field. See Chart 1 for a complete list of font names. Also note that the on-screen display of the bar code will not look correct. Only the data printed by the mail merge will be in the correct bar code 128 format.

## Bar Code 128 Character Tables (2 – 4)

In bar code 128, each specific bar code pattern can represent 3 different values to your bar code reader. *(That's what each subset of bar code 128 does.)* For example, in Subset A, the bar code pattern for a lower case "m" is read by your bar code reader as a carriage return. In Subset B, this same bar code pattern is read by your bar code reader as the letter "m". In Subset C, this same bar code pattern is read by your bar code reader as the numeric value 76. Tables 2 – 4 show what character on your keyboard *(ASCII: Character or Position)* will generate each specific data character from your bar code scanner *(in subsets: Code A, Code B, or Code C)*. Each table also shows the *Value* of each character, which is used in calculating the checksum.

You must use the specific table for the subset you are in. If you are using Subset A, Table 2 shows the character mapping. In subset A, lower case ASCII letters are mapped to control characters.  Subset B has a standard ASCII mapping, as shown in Table 3. In subset C, shown in Table 4, all ASCII character positions map to pairs of numbers from 00 – 99.
.
Finally, since bar code 128 has more bar code patterns than there are characters in the alphabet, some additional *(non-ASCII)* characters must be used. For example, when you need to enter a bar code pattern represented by one of these high ASCII characters, say the "Â", you can enter that character by hand by using the numeric keypad. Hold the Alt key down, press-and-release the 0, then the 1, then the 9, and then the 4. Finally release the Alt key. You have entered the character code 0194 this way. In Word you must use Insert, Symbol, and then select the specific character you need. If you are using a program to enter these characters, drop the leading 0, so a Visual Basic program would use Chr(194).

# Table 2,  Subset A

| ASCII Char | Pos | Code A | Value | ASCII Char | Pos | Code A | Value | ASCII Char | Pos | Code A | Value |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ä | 228 | Space | 00 | D | 68 | D | 36 | h | 104 | BS | 72 |
| ! | 33 | ! | 01 | E | 69 | E | 37 | i | 105 | HT | 73 |
| " | 34 | " | 02 | F | 70 | F | 38 | j | 106 | LF | 74 |
| # | 35 | # | 03 | G | 71 | G | 39 | k | 107 | VT | 75 |
| $ | 36 | $ | 04 | H | 72 | H | 40 | l | 108 | FF | 76 |
| % | 37 | % | 05 | I | 73 | I | 41 | m | 109 | CR | 77 |
| & | 38 | & | 06 | J | 74 | J | 42 | n | 110 | SO | 78 |
| ' | 39 | ' | 07 | K | 75 | K | 43 | o | 111 | SI | 79 |
| ( | 40 | ( | 08 | L | 76 | L | 44 | p | 112 | DLE | 80 |
| ) | 41 | ) | 09 | M | 77 | M | 45 | q | 113 | DC1 | 81 |
| * | 42 | * | 10 | N | 78 | N | 46 | r | 114 | DC2 | 82 |
| + | 43 | + | 11 | O | 79 | O | 47 | s | 115 | DC3 | 83 |
| , | 44 | , | 12 | P | 80 | P | 48 | t | 116 | DC4 | 84 |
| - | 45 | - | 13 | Q | 81 | Q | 49 | u | 117 | NAK | 85 |
| . | 46 | . | 14 | R | 82 | R | 50 | v | 118 | SYN | 86 |
| / | 47 | / | 15 | S | 83 | S | 51 | w | 119 | ETB | 87 |
| 0 | 48 | 0 | 16 | T | 84 | T | 52 | x | 120 | CAN | 88 |
| 1 | 49 | 1 | 17 | U | 85 | U | 53 | y | 121 | EM | 89 |
| 2 | 50 | 2 | 18 | V | 86 | V | 54 | z | 122 | SUB | 90 |
| 3 | 51 | 3 | 19 | W | 87 | W | 55 | Â | 194 | ESC | 91 |
| 4 | 52 | 4 | 20 | X | 88 | X | 56 | Ã | 195 | FS | 92 |
| 5 | 53 | 5 | 21 | Y | 89 | Y | 57 | Ä | 196 | GS | 93 |
| 6 | 54 | 6 | 22 | Z | 90 | Z | 58 | Å | 197 | RS | 94 |
| 7 | 55 | 7 | 23 | [ | 91 | [ | 59 | Æ | 198 | US | 95 |
| 8 | 56 | 8 | 24 | \ | 92 | \ | 60 | Ç | 199 | FNC3 | 96 |
| 9 | 57 | 9 | 25 | ] | 93 | ] | 61 | È | 200 | FNC2 | 97 |
| : | 58 | : | 26 | ^ | 94 | ^ | 62 | É | 201 | Shift | 98 |
| ; | 59 | ; | 27 | _ | 95 | _ | 63 | Ê | 202 | Code C | 99 |
| < | 60 | < | 28 | ` | 96 | NUL | 64 | Ë | 203 | Code B | 100 |
| = | 61 | = | 29 | a | 97 | SOH | 65 | Ì | 204 | FNC4 | 101 |
| > | 62 | > | 30 | b | 98 | STX | 66 | Í | 205 | FNC1 | 102 |
| ? | 63 | ? | 31 | c | 99 | ETX | 67 | â | 226 | " | 02 |
| @ | 64 | @ | 32 | d | 100 | EQT | 68 | { | 123 | **Start A** | 103 |
| A | 65 | A | 33 | e | 101 | ENQ | 69 | \| | 124 | **Start B** | 104 |
| B | 66 | B | 34 | f | 102 | ACK | 70 | } | 125 | **Start C** | 105 |
| C | 67 | C | 35 | g | 103 | BEL | 71 | ~ | 126 | **Stop** | |

# Table 3, Subset B

| ASCII Char | Pos | Code B | Value | | ASCII Char | Pos | Code B | Value | | ASCII Char | Pos | Code B | Value |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ä | 228 | Space | 00 | | D | 68 | D | 36 | | h | 104 | h | 72 |
| ! | 33 | ! | 01 | | E | 69 | E | 37 | | i | 105 | i | 73 |
| " | 34 | " | 02 | | F | 70 | F | 38 | | j | 106 | j | 74 |
| # | 35 | # | 03 | | G | 71 | G | 39 | | k | 107 | k | 75 |
| $ | 36 | $ | 04 | | H | 72 | H | 40 | | l | 108 | l | 76 |
| % | 37 | % | 05 | | I | 73 | I | 41 | | m | 109 | m | 77 |
| & | 38 | & | 06 | | J | 74 | J | 42 | | n | 110 | n | 78 |
| ' | 39 | ' | 07 | | K | 75 | K | 43 | | o | 111 | o | 79 |
| ( | 40 | ( | 08 | | L | 76 | L | 44 | | p | 112 | p | 80 |
| ) | 41 | ) | 09 | | M | 77 | M | 45 | | q | 113 | q | 81 |
| * | 42 | * | 10 | | N | 78 | N | 46 | | r | 114 | r | 82 |
| + | 43 | + | 11 | | O | 79 | O | 47 | | s | 115 | s | 83 |
| , | 44 | , | 12 | | P | 80 | P | 48 | | t | 116 | t | 84 |
| - | 45 | - | 13 | | Q | 81 | Q | 49 | | u | 117 | u | 85 |
| . | 46 | . | 14 | | R | 82 | R | 50 | | v | 118 | v | 86 |
| / | 47 | / | 15 | | S | 83 | S | 51 | | w | 119 | w | 87 |
| 0 | 48 | 0 | 16 | | T | 84 | T | 52 | | x | 120 | x | 88 |
| 1 | 49 | 1 | 17 | | U | 85 | U | 53 | | y | 121 | y | 89 |
| 2 | 50 | 2 | 18 | | V | 86 | V | 54 | | z | 122 | z | 90 |
| 3 | 51 | 3 | 19 | | W | 87 | W | 55 | | Â | 194 | { | 91 |
| 4 | 52 | 4 | 20 | | X | 88 | X | 56 | | Ã | 195 | \| | 92 |
| 5 | 53 | 5 | 21 | | Y | 89 | Y | 57 | | Ä | 196 | } | 93 |
| 6 | 54 | 6 | 22 | | Z | 90 | Z | 58 | | Å | 197 | ~ | 94 |
| 7 | 55 | 7 | 23 | | [ | 91 | [ | 59 | | Æ | 198 | DEL | 95 |
| 8 | 56 | 8 | 24 | | \ | 92 | \ | 60 | | Ç | 199 | FNC3 | 96 |
| 9 | 57 | 9 | 25 | | ] | 93 | ] | 61 | | È | 200 | FNC2 | 97 |
| : | 58 | : | 26 | | ^ | 94 | ^ | 62 | | É | 201 | Shift | 98 |
| ; | 59 | ; | 27 | | _ | 95 | _ | 63 | | Ê | 202 | Code C | 99 |
| < | 60 | < | 28 | | ` | 96 | ` | 64 | | Ë | 203 | FNC4 | 100 |
| = | 61 | = | 29 | | a | 97 | a | 65 | | Ì | 204 | Code A | 101 |
| > | 62 | > | 30 | | b | 98 | b | 66 | | Í | 205 | FNC1 | 102 |
| ? | 63 | ? | 31 | | c | 99 | c | 67 | | â | 226 | " | 02 |
| @ | 64 | @ | 32 | | d | 100 | d | 68 | | { | 123 | **Start A** | 103 |
| A | 65 | A | 33 | | e | 101 | e | 69 | | \| | 124 | **Start B** | 104 |
| B | 66 | B | 34 | | f | 102 | f | 70 | | } | 125 | **Start C** | 105 |
| C | 67 | C | 35 | | g | 103 | g | 71 | | ~ | 126 | **Stop** | |

# Table 4, Subset C

| ASCII Char | Pos | Code C | Value | | ASCII Char | Pos | Code C | Value | | ASCII Char | Pos | Code C | Value |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ! | 33 | 00 | 00 | | E | 69 | 36 | 36 | | i | 105 | 72 | 72 |
| " | 34 | 01 | 01 | | F | 70 | 37 | 37 | | j | 106 | 73 | 73 |
| # | 35 | 02 | 02 | | G | 71 | 38 | 38 | | k | 107 | 74 | 74 |
| $ | 36 | 03 | 03 | | H | 72 | 39 | 39 | | l | 108 | 75 | 75 |
| % | 37 | 04 | 04 | | I | 73 | 40 | 40 | | m | 109 | 76 | 76 |
| & | 38 | 05 | 05 | | J | 74 | 41 | 41 | | n | 110 | 77 | 77 |
| ' | 39 | 06 | 06 | | K | 75 | 42 | 42 | | o | 111 | 78 | 78 |
| ( | 40 | 07 | 07 | | L | 76 | 43 | 43 | | p | 112 | 79 | 79 |
| ) | 41 | 08 | 08 | | M | 77 | 44 | 44 | | q | 113 | 80 | 80 |
| * | 42 | 09 | 09 | | N | 78 | 45 | 45 | | r | 114 | 81 | 81 |
| + | 43 | 10 | 10 | | O | 79 | 46 | 46 | | s | 115 | 82 | 82 |
| , | 44 | 11 | 11 | | P | 80 | 47 | 47 | | t | 116 | 83 | 83 |
| - | 45 | 12 | 12 | | Q | 81 | 48 | 48 | | u | 117 | 84 | 84 |
| . | 46 | 13 | 13 | | R | 82 | 49 | 49 | | v | 118 | 85 | 85 |
| / | 47 | 14 | 14 | | S | 83 | 50 | 50 | | w | 119 | 86 | 86 |
| 0 | 48 | 15 | 15 | | T | 84 | 51 | 51 | | x | 120 | 87 | 87 |
| 1 | 49 | 16 | 16 | | U | 85 | 52 | 52 | | y | 121 | 88 | 88 |
| 2 | 50 | 17 | 17 | | V | 86 | 53 | 53 | | z | 122 | 89 | 89 |
| 3 | 51 | 18 | 18 | | W | 87 | 54 | 54 | | Â | 194 | 90 | 90 |
| 4 | 52 | 19 | 19 | | X | 88 | 55 | 55 | | Ã | 195 | 91 | 91 |
| 5 | 53 | 20 | 20 | | Y | 89 | 56 | 56 | | Ä | 196 | 92 | 92 |
| 6 | 54 | 21 | 21 | | Z | 90 | 57 | 57 | | Å | 197 | 93 | 93 |
| 7 | 55 | 22 | 22 | | [ | 91 | 58 | 58 | | Æ | 198 | 94 | 94 |
| 8 | 56 | 23 | 23 | | \ | 92 | 59 | 59 | | Ç | 199 | 95 | 95 |
| 9 | 57 | 24 | 24 | | ] | 93 | 60 | 60 | | È | 200 | 96 | 96 |
| : | 58 | 25 | 25 | | ^ | 94 | 61 | 61 | | É | 201 | 97 | 97 |
| ; | 59 | 26 | 26 | | _ | 95 | 62 | 62 | | Ê | 202 | 98 | 98 |
| < | 60 | 27 | 27 | | ` | 96 | 63 | 63 | | Ë | 203 | 99 | 99 |
| = | 61 | 28 | 28 | | a | 97 | 64 | 64 | | Ì | 204 | Code b | 100 |
| > | 62 | 29 | 29 | | b | 98 | 65 | 65 | | Í | 205 | Code a | 101 |
| ? | 63 | 30 | 30 | | c | 99 | 66 | 66 | | Î | 206 | FNC1 | 102 |
| @ | 64 | 31 | 31 | | d | 100 | 67 | 67 | | â | 226 | 01 | 01 |
| A | 65 | 32 | 32 | | e | 101 | 68 | 68 | | { | 123 | **Start A** | 103 |
| B | 66 | 33 | 33 | | f | 102 | 69 | 69 | | \| | 124 | **Start B** | 104 |
| C | 67 | 34 | 34 | | g | 103 | 70 | 70 | | } | 125 | **Start C** | 105 |
| D | 68 | 35 | 35 | | h | 104 | 71 | 71 | | ~ | 126 | **Stop** | |

# Glossary

**Application Program:** a computer program that performs useful work not related to the computer itself. Examples are word processors, spreadsheets, accounting systems, and engineering programs.

**Auto Sense:** A way to switch back and forth between subsets in Bar Code 128.to minimize the length of the bar code.

**ASCII:** American Standard Code for Information Interchange – a standard code for representing characters as numbers that is used on most microcomputers, computer terminals, and printers.

**Aspect ratio**: the ratio of height to width.

**Character string**:  a sequence of characters stored in a computer and treated as a single data item.

**Checksum:** a number that accompanies data transferred from one place to another and helps to ensure that the data was transferred correctly.

**Clipboard:** a holding area to which information can be copied in order to transfer it from one application to another.

**CPI:** characters per inch.

**DPI:** dots per inch: the number of pixels or printer dots per linear inch.

**Font:** a complete collection of characters, in a consistent style and size. This includes upper and lower case letters, numerals, punctuation, ligatures, and reference marks.

**Modulo:** a mathematical operation that gives the remainder when one number is divided by another.

**Pixels**: one of the individual dots that make up a graphical image.

**Radio button:** small circles in a dialog box, only one of which can be chosen at a time. Choosing any button with the mouse causes all the other buttons in the set to be cleared.

**Scalable**: able to be used on a large or small scale without major changes.

**Scalable font**: a font that can be used to print characters of any size. Many newer laser printers include scalable fonts.

**SCC-14:** Shipping Container Code- a bar code for fixed content shipping containers.

**SSCC-18:** Serial Shipping Container Code- a bar code for fixed content shipping containers that also includes tracking information.

**Symbology:** a method of representing information by printed characters.

**TrueType Font:** an outline typeface that can be scaled or sized to practically any size.

**Scanner:** a device that enables a computer to read printed or handwritten page.

**Start Code**: the leading character of a bar code that identifies what kind of bar code it is.

**Stop Code**: the last character of a bar code that tells the bar code reader when it has reached the end of the code.

**UPC:** Uniform Product Code- the 12 digit bar code used on all retail products in North America

**Utility**: a program that assists in the operation of a computer but does not do the main work for which the computer was bought.

**Visual Basic Function**: a small program written in Visual Basic and used in Microsoft products like Excel, Access, or Word to help create bar codes.