

ELFRING FONTS INC

BAR CODES AND MORE

This package includes five of the most popular bar codes in OpenType font format. You also get OCR-A and OCR-B fonts. Most bar codes come in three different aspect ratios. By switching between these fonts you can print bar codes with the same width but at different heights.

This package includes:

Bar Code 2/5: (0123456789)

Bar Code 2/5 Interleaved: { "8Ndz }

Bar Code 3/9: *0123456789*

UPC-A:

W35173|H@DIEf

Postnet Bar Code: (6018300615)

TABLE OF CONTENTS

Installation.....	2
Bar Code Basics.....	2
Bar Code 2/5.....	3
Bar Code 2/5 Interleaved.....	3
Bar Code 3/9.....	6
Postal Bar Code.....	8
UPC-A.....	8
BarMore Utility Program.....	11
Embedding Bar Code Fonts in PDF Files.....	12
Visual Basic Macro Functions.....	12
Glossary.....	14

INSTALLATION

Our Barcode.exe program will install this package and font set in your copy of Windows. The fonts will be installed into the proper Windows font folder automatically. Our bar code utility program, BarMore.exe, this 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 BarMore.exe utility program. To access the utility program or the documentation, click on Start, Programs, Bar Codes & More, and select the program or documentation item you want to view.

USER ACCESSIBLE FILES

In Windows 11 - 7 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 "BarMore" in each user's Documents area. You can find these files as follows:

Windows XP:	My Documents\BarMore
Windows 7:	Libraries\Documents\ BarMore
Windows 8:	Libraries\Documents\ BarMore
Windows 10:	Libraries\Documents\ BarMore
Windows 11:	Libraries\Documents\ BarMore

BAR CODE BASICS

A bar code is made up of a series of parallel vertical bars and spaces. Bar codes are designed to convey information in a machine-readable format. In addition, some bar codes include a human readable portion so that a person can easily understand the bar code. Bar codes always start with a special character, or symbol, that tells the machine scanning that bar code to start the reading process. *(This Start character will also tell the reader what bar code symbology is being used.)* A bar code always ends with a special character, or symbol, that tells the reader that this is the end of the bar code *(the Stop character)*.

Some bar codes also require a checksum. A checksum is a special character that is added to your bar code. The checksum tells the bar code reader that the bar code is correct. The checksum character is read by the bar code scanner, but it is not passed along as part of the data. The checksum must be printed after the data, and before the Stop code.

The bar codes in this package are implemented as OpenType fonts. Each character in the font corresponds to its matching bar code pattern. To use a bar code font, switch to the bar code font, enter the bar code Start character, the data that you want to encode, a possible checksum character, and the Stop character. Windows will display this data as a bar code on screen and will print a bar code on virtually any printer connected to Windows. For examples of how to do this, run the BarMore program.

OpenType fonts are scaled by changing their height. The character's width is altered proportionate to the change in height. You can pick virtually any font height by changing the point size. *(There are 72 points to the inch.)* Unfortunately, when printing bar codes you frequently want to control both the bar height and its width, independent of each other. *(Printers and scanners can only handle a fixed range of print densities.)*

To overcome this limitation, our scalable bar code font set includes three different aspect ratios of most bar codes. These different aspect ratios let you vary both the height of the characters and their width independent of each other. If you need thicker bars, which print at the same height, you use a shorter bar code font at a bigger point size. If you need thinner bars, which print at the same height, you use a taller bar code font at a smaller height. By switching between the three different versions of the same bar code font, you should be able to print these bar codes at both the height and the width you need.

BAR CODE 2/5

Bar Code 2/5 is a fairly old bar code format with a low information density. Bar Code 2/5 can only encode the numbers 0 - 9. No checksum is necessary, and a separate Start and Stop bar pattern must be used. The Start character is encoded as the (and the Stop Character is encoded as the). To print a readable bar code, you must add the Start character before your data and append a Stop character after your data. Maximum print density (*not including Start/Stop codes*) is 7.14 characters/inch on 300 dpi printers, or 14.28 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. These fonts appear in Windows font menus as:

<i>Typeface Name</i>	<i>Bar Code</i>
Bar Code 25 Short	(0 1 2 3 4 5) Thick bars
Bar Code 25	(0 1 2 3 4 5) Standard bars
Bar Code 25 Tall	(0 1 2 3 4 5) Thin bars

Example

To print a bar code of the data 012345, you need to add a Start character to the beginning of the string, and the Stop character at the end of the string. So the actual string to print is (012345). This becomes the bar code:

012345 → (012345) → (0 1 2 3 4 5)

BAR CODE 2/5 INTERLEAVED

These scalable Bar Code 2 of 5 interleaved fonts appear in Windows font menus as:

<i>Typeface Name</i>	<i>Bar Code</i>
Bar Code 25 Interleaved Short	{ — C } Thick bars
Bar Code 25 Interleaved	{ — C } Standard bars
Bar Code 25 Interleaved Tall	{ — C } Thin bars

Bar Code 2 of 5 is a high density, self-checking bar code format. It achieves this high density printing by assigning a single bar code pattern to every *pair* of digits from 00 through 99. Thus, bar code 2/5 interleaved can not represent a single digit number like a 4. It can only represent a

number with two digits, like 04. There are 102 bar code patterns in the font: a Start code, a Stop code, and 100 bar codes for the pairs of digits 00 through 99. *(You must use an even number of digits in your bar code. If the code contains an odd number of digits you must add a leading 0.)*

Maximum print density *(not including Start/Stop codes)* is 11.1 numbers/inch on 300 dpi printers, or 22.2 dpi on 600 dpi printers. All data for bar code 2 of 5 interleaved is encoded. To encode a number, say 27, you look up the number 27 in the table below. The ASCII code next to that number is the character that must be used with the bar code 2/5 font to represent that number pair. The following chart shows what ASCII position to use for each pair of digits.

Bar Code 2/5 Interleaved ASCII Position vs Number Pair

ASCII		Number	ASCII		Number	ASCII		Number	ASCII		Number
Pos	Char	Pair	Pos	Char	Pair	Pos	Char	Pair	Pos	Char	Pair
33	!	00	59	;	26	85	U	52	111	o	78
34	"	01	60	<	27	86	V	53	112	p	79
35	#	02	61	=	28	87	W	54	113	q	80
36	\$	03	62	>	29	88	X	55	114	r	81
37	%	04	63	?	30	89	Y	56	115	s	82
38	&	05	64	@	31	90	Z	57	116	t	83
39	'	06	65	A	32	91	[58	117	u	84
40	(07	66	B	33	92	\	59	118	v	85
41)	08	67	C	34	93]	60	119	w	86
42	*	09	68	D	35	94	^	61	120	x	87
43	+	10	69	E	36	95	_	62	121	y	88
44	,	11	70	F	37	96	`	63	122	z	89
45	-	12	71	G	38	97	a	64			
46	.	13	72	H	39	98	b	65	123	{	Start
47	/	14	73	I	40	99	c	66	125	}	Stop
48	0	15	74	J	41	100	d	67			
49	1	16	75	K	42	101	e	68	161	i	90
50	2	17	76	L	43	102	f	69	162	¢	91
51	3	18	77	M	44	103	g	70	163	£	92
52	4	19	78	N	45	104	h	71	164	¤	93
53	5	20	79	O	46	105	i	72	165	¥	94
54	6	21	80	P	47	106	j	73	166	¦	95
55	7	22	81	Q	48	107	k	74	167	§	96
56	8	23	82	R	49	108	l	75	168	¨	97
57	9	24	83	S	50	109	m	76	169	©	98
58	:	25	84	T	51	110	n	77	170	ª	99

How do you encode a number into the correct characters for our bar code 2/5 interleaved font? If you want to use Access, Excel, or Word to print bar codes, see the documentation on our Visual Basic functions later in this manual, or you can use our Bar25i program to do it automatically.

Example

To manually encode the data 01234567, you need to convert the numbers into pairs. So break the number 01234567 up into pairs of digits first.

01234567 = 01, 23, 45, 67

Now look up those pairs of numbers in Table 1 to get their ASCII equivalents.

01234567 = 01, 23, 45, 67 = "8Nd

Finally, add the Start character to the beginning of the string, and the Stop character at the end of the string.

01234567 = 01, 23, 45, 67 = "8Nd = {"8Nd}

So the actual string to print using our bar code 2 of 5 interleaved font is {"8Nd}. This becomes the bar code:

01234567 → 01, 23, 45, 67 → "8Nd → {"8Nd} → { " 8Nd }

To encode the number 96572510 you follow the same methods:

96572510 → 96, 57, 25, 10 → \$Z:+ → {\$Z:+} → { \$ Z : + }

BAR CODE 2/5 INTERLEAVED OPTIONAL CHECKSUM

Bar code 2/5 Interleaved does offer an optional checksum, which is rarely used. The checksum consists of a single digit, modulo 10 number. All bar code 2/5 interleaved codes must contain an even number of digits, since each bar code pattern encodes two digits. So if you want to add a checksum, which is a single digit, you must start with an *odd* number of data digits. To calculate a checksum:

- 1) Start with an odd number of data digits
- 2) Assign an alternating weighting factor of 3, 1, 3, 1, ... starting from the right or least significant digit.
- 3) Multiply each digit times its weighting factor (either a 3 or a 1) and sum the result.
- 4) The checksum digit is the number which, when added to the sum, makes the sum an even multiple of 10.

For example, if you want to encode the data "01234" (*note the odd number of digits*) you multiply the least significant digit by 3, the next by 1, ... and sum these products. The result is 22. The checksum digit will be an 8, since $8 + 22 = 30$, which is evenly divisible by 10. So the complete bar code data would be "012348".

Data	0	1	2	3	4	
Weights	3	1	3	1	3	
Product	0	1	6	3	12	= 22

01234 → 012348 → "8Q → {"8Q} → { " 8Q }

BAR CODE 3/9

This scalable Bar Code 3 of 9 font comes in three different aspect ratios. In addition, the set includes both standard and human readable versions of bar code 3 of 9. These fonts appear in Windows font menus as:

<i>Typeface Name</i>	<i>Bar Code</i>	
Bar Code 39 Short	*123ABC*	Thick bars
Bar Code 39	*123ABC*	Standard bars
Bar Code 39 Tall	*123ABC*	Thin bars

Maximum print density (*not including Start/Stop codes*) is 6.25 characters/inch on 300 dpi printers, or 12.5 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.

Bar code 3 of 9 does not require a checksum, but an optional checksum is possible. All bar code 3 of 9 bar codes use the * (*asterisk*) to start and to end a bar code. The bar code 3 of 9 format lets you encode numbers, capitol letters, some punctuation, and the space character in your bar code data. These fonts let you access the following characters as bar codes:

ASCII	Code 39		ASCII	Code 39		ASCII	Code 39		ASCII	Code 39
space	~		3	3		E	E		P	P
\$	\$		4	4		F	F		Q	Q
%	%		5	5		G	G		R	R
*	Start/Stop		6	6		H	H		S	S
+	+		7	7		I	I		T	T
-	-		8	8		J	J		U	U
.	.		9	9		K	K		V	V
/	/		A	A		L	L		W	W
0	0		B	B		M	M		X	X
1	1		C	C		N	N		Y	Y
2	2		D	D		O	O		Z	Z

Note that since OpenType fonts do not allow a printable character to be defined for the space position, we have mapped the actual bar code 3 of 9 space to the tilde ~ character. If you want to use Access, Excel, or Word to print bar codes, see the information on Visual Basic functions in this manual.

Example

To print a bar code of the data 012345, you need to add a Start character to the beginning of the string, and the matching Stop character at the end of the string. So the actual string to print is *012345*. This becomes the bar code:

012345 → *012345* → *012345* standard bar code 3 of 9

MICROSOFT WORD WARNING!

Bar codes always need a Start and a Stop character in order to be readable. In bar code 3/9, the Start and the Stop character are both the same symbol, the asterisk *. So all bar code 3/9 bar codes must begin with an * and end with an *. To bar code “12345”, you have to add those Start and Stop characters- “*12345*” to your data.

Unfortunately, if you are using Word, when you start text with an * and end text with an *, Word automatically removes the two * characters and bolds the text in between. Word turns the string “*12345*” into “**12345**”! This makes the bar code unreadable. You **must** turn the auto-bolding feature off in Word before you can use Word to create bar code 3/9. To turn this feature off in Word, click on: Tools, AutoCorrect, AutoFormat As You Type, and then click off the *Bold* box. Finally, click on the OK button.

BAR CODE 3/9 CHECKSUM

Bar code 3 of 9 supports an optional checksum. The checksum character follows your data and is positioned before the Stop code. The checksum is calculated by assigning each character a numerical value, summing the values of all characters, and then performing a modulo 43 division. The table below shows the numerical value assignment for bar code 3 of 9.

Code 39	Value		Code 39	Value		Code 39	Value		Code 39	Value
0	0		B	11		M	22		X	33
1	1		C	12		N	23		Y	34
2	2		D	13		O	24		Z	35
3	3		E	14		P	25		-	36
4	4		F	15		Q	26		.	37
5	5		G	16		R	27		space	38
6	6		H	17		S	28		\$	39
7	7		I	18		T	29		/	40
8	8		J	19		U	30		+	41
9	9		K	20		V	31		%	42
A	10		L	21		W	32			

For example, to generate a checksum for the string “CODE 39”, look up the Values of these characters and add them together:

$$12 + 24 + 13 + 14 + 38 + 3 + 9 = 113$$

Now divide by 43 and use the remainder (not the result) as the checksum:

$$113 / 43 = 2 \text{ remainder } 27$$

So the checksum has a value of 27, which from the chart above is an "R". Thus the string to use to encode a bar code 3/9 with checksum for "CODE 39" is

CODE~39R

CODE 39 → CODE~39 → CODE~39R → *CODE~39R* → *CODE~39R*

POSTAL BAR CODES

This scalable PostNet Bar Code is easy to use. You **must** select a point size of 18 when using it. Other sizes are not valid for Post Office use. If you have never used bar codes, you will probably need some pointers on how they work. A Postal Bar Code consists of 4 separate parts: a lead-in character, the zip code, a checksum, and a lead-out character.

Typeface Name	Bar Code
Postal	(601832) Standard 18 point

The Postal Bar Code font contains bar code patterns for the numbers 0 - 9, plus the lead-in and lead-out characters. Your checksum is printed using a digit from 0 - 9. To print a Postal Bar Code, (*ignoring the checksum for the moment*) switch to the Postal font, type the lead-in character "(", type the complete 5, 9, or 11 digit zip code, the checksum, and the lead-out character ")". Then switch back to your normal font. For example, to print the zip code 60605-1234 type:

606051234 → 606051234X → (606051234X) → (6060512343) Where "X" is the checksum.

CHECKSUMS

The checksum digit is easy to calculate. Basically, the sum of all the digits in the bar code PLUS the checksum digit must be a multiple of 10. (*The sum must be evenly divisible by 10.*) For example, take the zip code 12345-6789. The sum of the digits is 45 (1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9). So a checksum of 5 is need to make the total divisible by 10. (*Including the checksum the total sum is 50.*) So, to print this bar code type with the Postal Bar Code font and a point size of 18:

(1234567895) → (1234567895)

UPC-A

The UPC-A bar code is a fairly complicated bar code. UPC-A bar codes always have a Start character, 5 numbers, a guard bar, 5 more numbers, a checksum, and a Stop character.

UPC-A

P 71001 | @@FII e

start data guard data stop

There are two versions of the UPC-A bar code font in this package, a normal one and a short one. Each UPC-A OpenType soft font actually contains four different bar code patterns for each number from 0 through 9. The font also includes a special guard bar stored in the "|" character. UPC-A product codes are always broken down into four separate parts: product code, left 5 digits, rights 5 digits, and checksum. Each part uses a **different** portion of the bar code font.

Typeface Name *Bar Code*

UPC-A

P12345 | FGHI @e

48

point

UPC-A Short

P12345 | FGHI @e

48

point

Product Code	ASCII Char	Pos	Left Digits	ASCII Char	Pos	Right Digits	ASCII Char	Pos	Check Sum	ASCII Char	Pos
0	P	80	0	0	48	0	@	64	0	`	96
1	Q	81	1	1	49	1	A	65	1	a	97
2	R	82	2	2	50	2	B	66	2	b	98
3	S	83	3	3	51	3	C	67	3	c	99
4	T	84	4	4	52	4	D	68	4	d	100
5	U	85	5	5	53	5	E	69	5	e	101
6	V	86	6	6	54	6	F	70	6	f	102
7	W	87	7	7	55	7	G	71	7	g	103
8	X	88	8	8	56	8	H	72	8	h	104
9	Y	89	9	9	57	9	I	73	9	i	105

To make a UPC-A bar code for the product number 7 35173 80495, first calculate the check sum. To do this, take the entire string of 11 digits (*product code, 5 left digits, and 5 right digits*), and starting from the left, add each of the odd position digits. (Add the first, third, fifth, ...) Multiply the result of this process by three (3). Now starting from the left again, add all the even digits together (second, fourth, sixth,...). Add this result to the previous result that was multiplied by 3. Take the least significant digit of this number (*the remainder if you divided this number by 10*) and subtract it from 10. If the answer is 10, change it to a 0. This is the checksum. For our number 7 35173 80495 we calculate this as follows:

odd digits $7 + 5 + 7 + 8 + 4 + 5 = 36$, $36 * 3 = 108$

even digits $3 + 1 + 3 + 0 + 9 = 16$

odd + even = 108 + 16 = 124 least significant digit = 4, so checksum = 10 - 4 = 6

Next we need to change the string 7 35173 80495 6 to use the proper bar code patterns found inside the UPC-A soft font. The product code “7” must come from the Product Code ASCII section. There a “7” is represented by the letter ‘W’. (*Use the two leftmost columns of the table above.*) The left five digits happen to come out the same, “35173”. Next add a “|” character for the guard bar between halves of the UPC-A product code. The right 5 digits, “80495”, become the letters “H@DIE”. Finally, the checksum “6” is mapped to the letter “f”. So the string that corresponds to the UPC-A code 7 35173 80495 6 in this bar code font is “W35173|H@DIEf”. You must use a point size ranging from 48 to 72 points.

735173 80495 → 7 35173 80495 6 → W 35173 | [H@DIE](#) f → W35173|H@DIEf

W35173|H@DIEf

UPC-A

Short (48 points)

W35173|H@DIEf W

35173 | H@DIE f

UPC-A (48 pts)

Product code (P - Y)	Left 5 digits (0 - 9)	Guard bar 	Right 5 digits (@ - I)	Checksum (‘ - i)
----------------------------	--------------------------	------------------	---------------------------	---------------------

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 you 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 add-on font set.

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 and More, How to Print Labels* to launch the file in your pdf viewer, or open the file printlabels.pdf with your pdf viewer (in the folder: My Documents\BarMore).

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 a folder called BarMore under your My Documents folder. (**My Documents\BarMore**) You can use the **Print Labels** button in our BarMore 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 “My Documents” folder. Look for a subfolder with the name: \BarMore.

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 BarMore 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 My Documents/BarMore folder with the file name “custom.rtf”. Our BarMore utility can directly access that template to let you easily print labels.

BARMORE UTILITY PROGRAM

The BarMore utility program was automatically added to your system when you ran our install program. You can access the BarMore utility from the Desktop icon, or by clicking on: Start, Programs, Elfring Bar Code More, Utility program. This program lets you enter data and build a bar

code from it. You can print a sample page of the resulting bar code, or transfer that bar code to another Windows program.

The BarMore program lets you select the type of bar code you want to use from a set of radio boxes at the top left portion of the screen. You can also select a font size, where applicable. Once you select a bar code type, any options for that particular bar code (*like an optional checksum*) will appear below it. The input box lets you enter any text you want to turn into a bar code. It is initially blank.

Once you have entered the text you want to encode, click on the **“Make”** button. The window below your text will show the actual string you need to generate a bar code with our fonts. The bottom window will show an actual bar code sample. You can copy that bar code string to the clipboard, by clicking on the **“Copy”** button. Once this data is in the clipboard, you can paste it into any Windows application. If you want to print bar code labels you can use the **“Labels”** button to select a label size and then open the label template in your word processor. The final button, **“Sample”** will print a copy of that bar code in a number of different point sizes.



If you try to enter a character that is not supported by the bar code type you are using, the BarMore program will display an error message.

Finally, if you have an active internet connection, the BarMore utility can check to see if a newer version of the program has been released. Just click on the **“Update”** button in the lower right corner of the program Window, or the version number in the About box. Your internet browser will start and a web page will tell you whether or not you have the latest version of the program.

Visual Basic FUNCTIONS FOR ACCESS, EXCEL, AND WORD

This package contains macros / functions that let you automatically build bar code strings in Excel, Access, and indirectly in Word. This set includes the following functions:

Function	Details
Bar25(Numbers)	Converts the input numeric data into a complete bar code 2/5. The function adds the Start code, appends the data, and then puts the Stop code at the end. Non-numeric data is ignored. This result must be formatted with one of the following typefaces: Bar Code25, Bar Code25Short, Bar Code25Tall.
Bar25I(Numbers)	Converts the input numeric data into a complete bar code 2/5 interleaved. The function adds the Start code, breaks the data up into number pairs and converts them to single characters, and then puts the Stop code at the end. Non-numeric data is ignored. You must enter an even number of digits, or a leading zero will be added. This result must be formatted with one of the following typefaces: Bar Code25 Interleave, Bar Code25 InterleaveShort, Bar Code25 InterleaveTall.
Bar25Ics(Numbers)	Converts the input numeric data into a complete bar code 2/5 interleaved. The function adds the Start code, breaks the data up into number pairs and converts them to single characters, adds the checksum digit and then puts the Stop code at the end. Non-numeric data is ignored. You must enter an odd number of digits, or a leading zero will be added. This result must be formatted with one of the following typefaces: Bar Code25 Interleave, Bar Code25 InterleaveShort, Bar Code25 InterleaveTall.
Bar39(Text)	Converts the input text data into a complete bar code 3/9. The function adds the Start code, throws away all characters that aren't in standard bar code 3/9, appends the data, and puts the Stop code at the end. This result must be formatted with one of the following typefaces: Bar Code 39, Bar Code 39 Short, Bar Code 39 Tall.
Bar39cs(Text)	Converts the input text data into a bar code 3/9 with checksum. The function adds the Start code, throws away all characters that aren't in standard bar code 3/9, appends the data, calculates and appends the checksum, and puts the Stop code at the end. This result must be formatted with one of the following typefaces: Bar Code 39, Bar Code 39 Short, Bar Code 39 Tall.
PostNet(Numbers)	Converts the input numeric data into a PostNet bar code with checksum. The function adds the Start code, throws away all non-numeric characters, appends the data, calculates and

	appends the checksum, and puts the Stop code at the end. This result must be formatted with the Postal typeface at 18 points.
BarUPC(Numbers)	Converts the input 11 or 12 digit number into a complete UPC-A bar code compatible with our UPC-A fonts. The function adds the Start code, translates and appends the data plus guard bar, calculates and adds the checksum, and puts the Stop code at the end. Non-numeric data is ignored, as are numbers greater than 12 digits in length. This result must be formatted with one of the following typefaces: UPC-A, UPC-A Short

Warning!

Unlike our utility program, *BarMore*, 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 *BarMoreE* utility program to verify that your data is correct before using these Visual Basic functions to mass produce bar codes

Using Visual Basic Functions in Excel

Open the spread sheet where you want to add bar code functions (*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 Codes and More package (*probably BarMore*). The Visual Basic file, **VBbarmor.bas**, should appear there. Select this file and open it. This will add a new function module, EFBARMOR, to your spread sheet. This module is automatically saved along with your spread sheet and adds the seven functions (*see Table 8*) to 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, =BarUPC(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 Table 1 for applicable font names.

Using Visual Basic functions in Access

Open the database where you want to add bar code functions (*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 Codes and More package (*probably BarMore*). The Visual Basic file, **VBbarmor.bas**, should appear there. Select this file and open it. This will add a new function module, EFBARMOR, to your database. This module is automatically saved along with your database and adds the seven functions (*see Table 8*) to it.

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 =BarUPC ([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 Table 1 for a complete listing of font names.

Using Visual Basic functions in Word

While Word does have the ability to use Visual Basic functions in macros, you need to be a macro expert to use these functions. You can, however, combine Word and Excel to print bar codes! 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. Now 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 Table 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 format.

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.

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 the bar code height to bar code 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: the number of characters a fixed width font will print 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.

Parity pattern: a clever way of encoding extra data, like a checksum, in the individual characters of a bar code.

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.

Symbology: a method of representing information by printed characters.

OpenType 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.

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

Copyright 2022 by Elfring Fonts, Inc.
4N899 W Mary Dr, St Charles, IL 60175
Phone: 1-630-440-2456

www.barcodingfonts.com