BAR CODES PLUS

This package contains TrueType font versions of all of the most popular bar codes formats plus OCR-A and OCR-B fonts. It also includes a Windows utility, BarPlus, which helps you make bar codes, and VBA macros for Excel, Access, and Word. Most of these bar code fonts come in multiple 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 Interleaved: {"8Ndz}

Bar Code 3/9: *012345ABCD*

Bar Code 93: (01234ABCDTK)

Bar Code 128: |12345ABCDª~

Bookland: 

Codabar: a12345c

EAN 8: 

EAN 13: 

Postnet: (6018300615)

UPC-A: 

UPC-E: [RCDUVG]5

INSTALLATION

Please use our Brcdplus.exe program to install these TrueType fonts in Windows. These fonts will be installed into Windows and automatically backed up in a folder named \barplus, *(unless you changed the folder name during installation).* The bar code utility program, BarPlus, will also be installed.

That's all there is to installing TrueType fonts in Windows. The bar code fonts you just installed will show up in all font menus in your Windows application programs. You can select any one of these bar code types just as you normally select any other font in your application. Windows will also display matching screen images of each character in the bar code font. For help in building bar codes, run the Bar25i utility program.

BAR CODE BASICS

A bar code is made up of a series of parallel vertical bars and spaces. Bar codes are designed to quickly convey information in a machine-readable format. Some bar codes include a human readable portion so that a person can also read the bar code. Bar codes always begin 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 TrueType fonts. Each character in the font corresponds to a 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 BarPlus program.

TrueType fonts are scaled by changing their height. A character's width is altered proportionate to a 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.)*

This package gives you numerous different print densities, or versions, of each bar code font. 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 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.

123 *123* *123*

To get an idea of how to use these bar code fonts, try running our utility program, BarPlus.exe (click on Start, Programs, Elfring Bar Codes Plus, Utility). If you are using current versions of Excel, Access, or Word, our VBA macro functions will let you use these fonts automatically. See the end of this manual for details on both the utility program and the VBA macros.

BAR CODE 2/5 INTERLEAVED

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

Chart 1

Typeface Name	*Bar Code*	
Bar 2/5i b	{ −C }	Thick bars (1/2 standard height)
Bar 2/5i b HR	{ −C }	Thick bars, human readable
Bar 2/5i c	{ −C }	Slightly thick bars
Bar 2/5i c HR	{ −C }	Slightly thick bars, human readable
Bar 2/5i d	{ −C }	Standard width bars
Bar 2/5i d HR	{ −C }	Standard width bars, human readable
Bar 2/5i e	{ −C }	Thinner width bars
Bar 2/5i e HR	{ −C }	Thinner width bars, human readable
Bar 2/5i f	{ −C }	Very thin bars (2X standard height)

Bar 2/5i f HR { −C } Very thin bars, human readable

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 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.)*

All data for bar code 2 of 5 interleaved is encoded. To encode a number, say 27, look up the number 27 in Table 1. 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. Table 1 shows what ASCII position to use for each pair of digits.

Table 1
Bar Code 2/5 Interleaved ASCII Position vs Number Pair

ASCII Pos	Char	Number Pair		ASCII Pos	Char	Number Pair		ASCII Pos	Char	Number Pair		ASCII Pos	Char	Number 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 / 194	¡ or Â	90
50	2	17		76	L	43		102	f	69		162 / 195	¢ or Ã	91
51	3	18		77	M	44		103	g	70		163 / 196	£ or Ä	92
52	4	19		78	N	45		104	h	71		164 / 197	¤ or Å	93
53	5	20		79	O	46		105	i	72		165 / 198	¥ or Æ	94
54	6	21		80	P	47		106	j	73		166 / 199	¦ or Ç	95
55	7	22		81	Q	48		107	k	74		167 / 200	§ or È	96
56	8	23		82	R	49		108	l	75		168 / 201	¨ or É	97
57	9	24		83	S	50		109	m	76		169 / 202	© or Ê	98
58	:	25		84	T	51		110	n	77		170 / 203	ª or Ë	99

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

To manually encode the data 01234567, you need to convert the numbers into pairs of digits:

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} = {"8Nd} = { " 8 N d }

To encode the number 96572510 you follow the same methods:

96572510 = 96, 57, 25, 10 = §Z:+ = {§Z:+} = { § Z : + }

BAR CODE 2/5 INTERLEAVED 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. Since all bar code 2/5 interleaved codes must contain an even number of digits, 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

ASPECT RATIOS

The wide to narrow ratio for these bar code fonts is 3.0. The only way to vary this ratio is to purchase a different bar code set. This bar code 2/5 interleaved font set includes 5 different aspect ratios of each bar code: .5 *(b),* .75 *(c),* 1.0 *(d),* 1.5 *(e),* and 2.0 *(f). (See Chart 1.)* 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 the a, b, c, d or e version. The "b" version will print a bar code of exactly the same length, but at .5 times the bar height. The "f" version will print a bar code of exactly the same length, but at 2 times the bar height. 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.

BAR CODE 3/9

This scalable Bar Code 3 of 9 font comes in six 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:

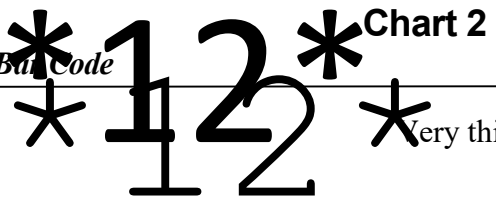


Chart 2

Typeface Name	*Bar Code*	
Bar Code 3/9 a		Very thick bars
Bar Code 3/9 a HR		Very thick bars, human readable
Bar Code 3/9 b	*12*	Thicker bars
Bar Code 3/9 b HR	*12*	Thicker bars, human readable
Bar Code 3/9 c	*12*	Slightly thicker bars
Bar Code 3/9 c HR	*12*	Slightly thicker bars, human readable
Bar Code 3/9 d	*12*	Standard width bars
Bar Code 3/9 d HR	*12*	Standard width bars, human readable
Bar Code 3/9 e	*12*	Thinner width bars
Bar Code 3/9 e HR	*12*	Thinner width bars, human readable
Bar Code 3/9 f	*12*	Very thin bars
Bar Code 3/9 f HR	*12*	Very thin bars, human readable

These different aspect ratios let you vary both the height of the characters and their width, or print density, independently of each other.

Bar Code 3 of 9 is a simple bar code format. It does not require a checksum, but an optional checksum can be used in special situations. All 3 of 9 bar codes use the * *(asterisk)* to start and end a bar code. The 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 characters shown in Table 2:

Table 2

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 TrueType 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 VBA functions later 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* standard bar code 3 of 9

012345 *012345* human readable 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.

ASPECT RATIOS

The wide to narrow ratio for these bar code fonts is 3.0. The only way to vary this ratio is to purchase a different bar code set. This bar code 3/9 font set includes 6 different aspect ratios of each bar code: .25 *(a),* .5 *(b),* .75 *(c),* 1.0 *(d),* 1.5 *(e),* and 2.0 *(f). (See Chart 2.)* 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 6.25 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 3/9 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. Table 3 shows the numerical Value assignment for bar code 3 of 9.

Table 3

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”, you 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 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

BAR CODE 3/9 FULL ASCII

Bar code 3 of 9 has only 43 characters defined for it. However, it is possible to encode all 128 ASCII characters using the Full ASCII mode of bar code 3 of 9. **Note** that Full ASCII is a function of the bar code reader, not the bar code. Your bar code scanner must have this feature and you must enable it in your reader before trying to scan Full ASCII data.

When a bar code reader scans Full ASCII data, it uses the symbols $ / % and + to modify the meaning of the other characters in bar code 3 of 9. So it takes two characters to create each new ASCII character. You get a much lower print density, but you will have the ability to print the full ASCII character set. Table 4 shows the encoding used to generate Full ASCII. **Note** that if your bar code reader does not support Full ASCII mode, you will simply get the two characters you used to represent the single ASCII character.

Table 4

ASCII Char	Decimal	Code 39	ASCII Char	Decimal	Code 39	ASCII Char	Decimal	Code 39	ASCII Char	Decimal	Code 39
NUL	00	%U	Space	32	~	@	64	%V	`	96	%W
SOH	01	$A	!	33	/A	A	65	A	a	97	+A
STX	02	$B	“	34	/B	B	66	B	b	98	+B
ETX	03	$C	#	35	/C	C	67	C	c	99	+C
EOT	04	$D	$	36	/D	D	68	D	d	100	+D
ENQ	05	$E	%	37	/E	E	69	E	e	101	+E
ACK	06	$F	&	38	/F	F	70	F	f	102	+F
BEL	07	$G	‘	39	/G	G	71	G	g	103	+G
BS	08	$H	(	40	/H	H	72	H	h	104	+H
HT	09	$I	)	41	/I	I	73	I	i	105	+I
LF	10	$J	*	42	/J	J	74	J	j	106	+J
VT	11	$K	+	43	/K	K	75	K	k	107	+K
FF	12	$L	,	44	/L	L	76	L	l	108	+L
CR	13	$M	-	45	-	M	77	M	m	109	+M
SO	14	$N	.	46	.	N	78	N	n	110	+N
SI	15	$O	/	47	/O	O	79	O	o	111	+O
DLE	16	$P	0	48	0	P	80	P	p	112	+P
DC1	17	$Q	1	49	1	Q	81	Q	q	113	+Q
DC2	18	$R	2	50	2	R	82	R	r	114	+R
DC3	19	$S	3	51	3	S	83	S	s	115	+S
DC4	20	$T	4	52	4	T	84	T	t	116	+T
NAK	21	$U	5	53	5	U	85	U	u	117	+U
SYN	22	$V	6	54	6	V	86	V	v	118	+V
ETB	23	$W	7	55	7	W	87	W	w	119	+W
CAN	24	$X	8	56	8	X	88	X	x	120	+X
EM	25	$Y	9	57	9	Y	89	Y	y	121	+Y
SUB	26	$Z	:	58	/Z	Z	90	Z	Z	122	+Z
ESC	27	%A	;	59	%F	[	91	%K	{	123	%P
FS	28	%B	<	60	%G	\	92	%L	|	124	%Q
GS	29	%C	=	61	%H	]	93	%M	}	125	%R
RS	30	%D	>	62	%I	^	94	%N	~	126	%S
US	31	%E	?	63	%J	_	95	%O	DEL	127	%T

BAR CODE 93

This scalable bar code 93 font set comes in six different aspect ratios. In addition, the set includes both standard and human readable versions of bar code 93. These fonts appear in Windows font menus as:

Chart 3

Typeface Name	*Bar Code*	
Code 93 a	(1234K3)	Very thick bars
Code 93 a HR	(1234K3)	Very thick bars, human readable
Code 93 b	(1234K3)	Thicker bars
Code 93 b HR	(1234K3)	Thicker bars, human readable
Code 93 c	(1234K3)	Slightly thicker bars
Code 93 c HR	(1234K3)	Slightly thicker bars, human readable
Code 93 d	(1234K3)	Standard width bars
Code 93 d HR	(1234K3)	Standard width bars, human readable

Code 93 e	(1234K3)	Thinner width bars
Code 93 e HR	(1234K3)	Thinner width bars, human readable
Code 93 f	(1234K3)	Very thin bars
Code 93 f HR	(1234K3)	Very thin bars, human readable

These different aspect ratios let you vary both the height of the characters and their width, or print density, independently of each other.

Bar code 93 is similar to bar code 3 of 9 in format. However, bar code 93 requires two independent checksums. Our bar code 93 uses the "(" character as a Start code and the ")" character as a Stop code. The bar code 93 format lets you encode numbers, capitol letters, some punctuation, and the space character in your bar code data. It also supports a Full ASCII mode, as detailed later. These fonts let you access the following characters as bar codes:

Table 5

ASCII Pos	ASCII Char	Code 93	ASCII Pos	ASCII Char	Code 93	ASCII Pos	ASCII Char	Code 93
32	space	~	66	B	B	84	T	T
36	$	$	67	C	C	85	U	U
37	%	%	68	D	D	86	V	V
43	+	+	69	E	E	87	W	W
45	-	-	70	F	F	88	X	X
46	.	.	71	G	G	89	Y	Y
47	/	/	72	H	H	90	Z	Z
48	0	0	73	I	I			
49	1	1	74	J	J	40	(	Start
50	2	2	75	K	K	41	)	Stop
51	3	3	76	L	L			
52	4	4	77	M	M	60	<	<
53	5	5	78	N	N	61	=	=
54	6	6	79	O	O	62	>	>
55	7	7	80	P	P	63	?	?
56	8	8	81	Q	Q			
57	9	9	82	R	R			
65	A	A	83	S	S			

Note that since TrueType fonts do not allow a printable character to be defined for the space position, we have mapped the actual bar code 93 space character to the tilde "~" character.

To print a bar code of the data 456ABC, you need to add the Start code "(" to the beginning of the string, calculate the two checksums *(see the next section)* "</" and append the checksum characters to the end of the string, and then add a Stop code at the very end of the string, ")".

"(" + "456ABC" + "</" + ")"

So the actual string to print is (ABCDE</). This becomes the bar code:

(456ABC</) (456ABC</) standard bar code 93

(456ABC</) (456ABC</) human readable bar code 93

BAR CODE 93 CHECKSUM

Bar code 93 requires a two character checksum. The checksum characters follow your data and are positioned before the Stop code. Each checksum is calculated with a weighted modulo 47 routine. If you want to use Access, Excel, or Word to print bar codes, see our VBA function documentation near the end of this manual. To calculate a checksum, each bar code 93 character is assigned a numerical value, from 0 to 46, shown in Table 6. This value is multiplied by a weighting factor and the result is summed. A modulo 47 division is then performed. The table below shows the numerical value assignment for bar code 93.

Table 6

Code 93	Value		Code 93	Value		Code 93	Value		Code 93	Value
0	0		C	12		O	24		-	36
1	1		D	13		P	25		.	37
2	2		E	14		Q	26		space	38
3	3		F	15		R	27		$	39
4	4		G	16		S	28		/	40
5	5		H	17		T	29		+	41
6	6		I	18		U	30		%	42
7	7		J	19		V	31		< (<)	43
8	8		K	20		W	32		= (=)	44
9	9		L	21		X	33		> (>)	45
A	10		M	22		Y	34		? (?)	46
B	11		N	23		Z	35			

To generate the checksum digits “C” and “K” for the string “456ABC”, you must first translate the characters to their values using Table 6. Next assign a weighting factor for the C checksum from right to left, starting at 1. *(The C weighting factor ranges from 1 to 20, then resets to 1, ranges up to 20 again, etc. The K weighting factor ranges from 1 to 15, and then resets to 1, ranges up to 15 again, etc.)* Now multiply each weighting factor times the data value and add all these numbers together. Finally, divide the sum by 47. The remainder of this division (not the result, the remainder) is the value of the C checksum. Remember that the C weight only ranges from 1 to 20.

```
Data        4   5   6   A   B   C   C   K
Data Value  4   5   6   10  11  12
C Weight            6   5   4   3   2   1
Product    24  25  24  30  22  12
```

To calculate the C checksum, multiply each weight factor times the data value and sum them, as follows:

(4 * 6) + (5 * 5) + (6 * 4) + (10 * 3) + (11 * 2) + (12 * 1) = 137

Then divide 137 by 47 , which equals 2 remainder 43. The checksum is the character with the value of the remainder, 43 = “<” (which is the “<” character- see Table 6).

The K checksum is done in a similar fashion. Remember that the K weight only ranges from 1 to 15.

```
Data        4   5   6   A   B   C   <   K
Data Value  4   5   6   10  11  12  43
K Weight    7   6   5   4   3   2   1
Product     28  30  30  40  33  24  43
```

(4 * 7) + (5 * 6) + (6 * 5) + (10 * 4) + (11 * 3) + (12 * 2) + (43 * 1) = 228

Divide the 228 by 47 and you get 4 remainder 40, so the K checksum value is 40, which is the "/" character. In this example the string to use with our bar code 93 fonts is:

(456ABC</) = (456ABC</)

Note that the *C Weight* value cycles from 1 to 20, and the *K Weight* value cycles from 1 to 15. When a weight value reaches its maximum, it automatically resets to 1 for the next item in line. This only matters on bar codes that are 14 digits or longer. So *C Weight* looks like: 1, 2, 3, … 19, 20, 1, 2, 3, … and the *K Weight* looks like: 1, 2, 3, … 14, 15, 1, 2, 3, ….

To calculate the checksum for the bar code data "BAR 5577" we follow the same process:

Data	B	A	R		5	5	7	7	C	K
Data Value	11	10	27	38	5	5	7	7	2	0
C Weight	8	7	6	5	4	3	2	1		
K Weight	9	8	7	6	5	4	3	2	1	

C = (11*8) + (10*7) + (27*6) + (38*5) + (5*4) + (5*3) + (7*2) + (7*1) = 566/47 = 12 remainder 2, so C = 2, which is the "2".

K = (11*9) + (10*8) + (27*7) + (38*6) + (5*5) + (5*4) + (7*3) + (7*2) + (2*1) = 678/47 = 14 remainder 20, so K = 20, which is the "K". **Note** that the space character must be replaced by the "~". The string to use with our bar code 93 fonts is:

(BAR~55772K) (BAR~55772K)

BAR CODE 93 FULL ASCII

Bar code 93 has only 47 characters defined for it. Four of those characters are reserved for encoding all 128 ASCII characters using the Full ASCII feature of bar code 93. Since these are reserved characters that are not part of the normal alphabet, code 93 provides non-ambiguous full ASCII, unlike bar code 3 of 9. *(You don't have to tell your bar code reader you want to read full ASCII.)*

Special symbols <, =, >, ? modify the meaning of all the other characters in bar code 93. Thus it takes two characters to create each new ASCII character. You get a lower print density, but you can print the entire ASCII character set. Table 7 shows the encoding used to generate Full ASCII. **Note** that you use < to print a <, = to print a =, > to print a >, and ? to print a ?.

To print the pair of letters "Aa" in bar code 93, you need to know which characters are in the standard code 93 symbology. *(See Table 5.)* Since a lower case "a" is not part of bar code 93, we need to look up a pair of characters to represent the "a". Table 7 shows that an "a" can be represented by "?A". When your bar code reader sees an "?A" it returns an "a". So to bar code "Aa" we use:

(A?A~1) = (A?A~1) = (A?A~1)

Table 7

ASCII Pos	ASCII Char	Code 93	ASCII Pos	ASCII Char	Code 93	ASCII Pos	ASCII Char	Code 93	ASCII Pos	ASCII Char	Code 93
0	NUL	=U	32	Space	~	64	@	=V	96	‘	=W
1	SOH	<A	33	!	>A	65	A	A	97	a	?A

2	STX	<B		34	“	>B		66	B	B		98	b	?B
3	ETX	<C		35	#	>C		67	C	C		99	c	?C
4	EOT	<D		36	$	$		68	D	D		100	d	?D
5	ENQ	<E		37	%	%		69	E	E		101	e	?E
6	ACK	<F		38	&	>F		70	F	F		102	f	?F
7	BEL	<G		39	‘	>G		71	G	G		103	g	?G
8	BS	<H		40	(	>H		72	H	H		104	h	?H
9	HT	<I		41	)	>I		73	I	I		105	i	?I
10	LF	<J		42	*	>J		74	J	J		106	j	?J
11	VT	<K		43	+	+		75	K	K		107	k	?K
12	FF	<L		44	,	>L		76	L	L		108	l	?L
13	CR	<M		45	-	-		77	M	M		109	m	?M
14	SO	<N		46	.	.		78	N	N		110	n	?N
15	SI	<O		47	/	/		79	O	O		111	o	?O
16	DLE	<P		48	0	0		80	P	P		112	p	?P
17	DC1	<Q		49	1	1		81	Q	Q		113	q	?Q
18	DC2	<R		50	2	2		82	R	R		114	r	?R
19	DC3	<S		51	3	3		83	S	S		115	s	?S
20	DC4	<T		52	4	4		84	T	T		116	t	?T
21	NAK	<U		53	5	5		85	U	U		117	u	?U
22	SYN	<V		54	6	6		86	V	V		118	v	?V
23	ETB	<W		55	7	7		87	W	W		119	w	?W
24	CAN	<X		56	8	8		88	X	X		120	x	?X
25	EM	<Y		57	9	9		89	Y	Y		121	y	?Y
26	SUB	<Z		58	:	>Z		90	Z	Z		122	z	?Z
27	ESC	=A		59	;	=F		91	[	=K		123	{	=P
28	FS	=B		60	<	=G		92	\	=L		124	|	=Q
29	GS	=C		61	=	=H		93	]	=M		125	}	=R
30	RS	=D		62	>	=I		94	^	=N		126	~	=S
31	US	=E		63	?	=J		95	_	=O		127	DEL	=T

ASPECT RATIOS

This complete bar code 93 font set includes 6 different aspect ratios of each bar code: .25 *(a),* .5 *(b),* .75 *(c),* 1.0 *(d),* 1.25 *(e),* and 1.5 *(f). (See Chart 3.)* 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 1.25 times the bar height. Maximum print density *(not including Start/Stop codes)* is 11.11 characters/inch on 300 dpi printers, or 22.22 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 128

This package contains 2 different versions of the bar code 128 font, one for subsets A & B, and a second for subset C. Each version has 3 separate variations, to let you control both bar code height and width *(or the aspect ratio)* separate of each other. Each variation comes in both standard and human readable formats. These fonts include:

Chart 4 Subsets A & B

Typeface Name ***Bar Code***

Code 128AB Tall	{CODEä128t~
Code 128AB	{CODEä128t~
Code 128AB Short	{CODEä128t~
Code 128AB HR Tall	{CODEä128t~
Code 128AB HR	{CODEä128t~
Code 128AB HR Short	{CODEä128t~

Chart 4 Subset C

Typeface Name	*Bar Code*
CCode 128C Tall	}-CYoP~
Code 128C	}-CYoP~
Code 128C Short	}-CYoP~
Code 128C HR Tall	}-CYoP~
Code 128C HR	}-CYoP~
Code 128C HR Short	}-CYoP~

Note that bar code 128 needs more characters *(106)* than are available in the standard ASCII character set *(character positions 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 have mapped 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 8-10, 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.

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 VBA function documentation starting on page 6. **Note** that Tables 8-10 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 8-10.
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

{CODE®128t~ - Bar code 128 A for "CODE 128"

{CODE®128t~ - 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	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		

}-CYoP~ - 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		

}²-8MYD~ - 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		

ASPECT RATIOS

This bar code 128 font set includes 3 different aspect ratios of each bar code: .5 *(short),* 1.0 *(normal),* and 1.5 *(tall). (See Chart 4.)* When you are building a bar code, start with the Normal 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 the Short or the Tall version. The short version will print a bar code of exactly the same length, but at .5 times the bar height. The tall version will print a bar code of exactly the same length, 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 13.6 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 128 CHARACTER TABLES (8 – 10)

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 8-10 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 8 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 9. In subset C, shown in Table 10, 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. A Delphi or Visual Basic program might use Chr(194).

Table 8, Bar Code 128 Subset A

ASCII Char	ASCII Pos	Code A	Value		ASCII Char	ASCII Pos	Code A	Value		ASCII Char	ASCII Pos	Code A	Value
ä	2	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 9, Bar Code 128 Subset B

ASCII Char	ASCII Pos	Code B	Value		ASCII Char	ASCII Pos	Code B	Value		ASCII Char	ASCII Pos	Code B	Value
ä	2	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	

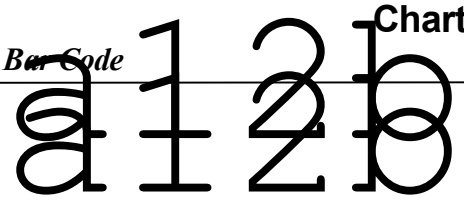
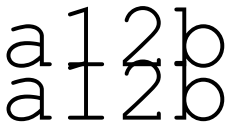
Table 10, Bar Code 128 Subset C

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	

RATIONALIZED CODABAR

This scalable Codabar font set comes in six different aspect ratios. In addition, the set includes both standard and human readable versions of Codabar. These fonts appear in Windows font menus as:

Chart 5

Typeface Name	*Bar Code*	
Codabar a		Very thick bars
Codabar a HR		Very thick bars, human readable
Codabar b		Thicker bars
Codabar b HR		Thicker bars, human readable
Codabar c	a12b	Slightly thicker bars
Codabar c HR	a12b	Slightly thicker bars, human readable
Codabar d	a12b	Standard width bars
Codabar d HR	a12b	Standard width bars, human readable
Codabar e	a12b	Thinner width bars
Codabar e HR	a12b	Thinner width bars, human readable
Codabar f	a12b	Very thin bars
Codabar f HR	a12b	Very thin bars, human readable

These different aspect ratios let you vary both the height of the characters and their width, or print density, independently of each other.

Codabar is a simple bar code format. It does not require a checksum, and you can only represent the characters shown in Table 11. The characters A, B, C, and/or D *(or a, b, c, & d),* function as both Start and Stop codes. All Codabar bar codes must start with one of the letters A-D and end with one of the letters A-D. Codabar bar codes can vary in length.

Table 11

ASCII		ASCII
0		-
1		$
2		:
3		/
4		.
5		+
6		A
7		B
8		C
9		D

To print a bar code of the data 012345, you need to add a Start character to the beginning of the string, and a Stop character at the end of the string. Depending on your system, you can use any character from A-D to start the bar code and any character from A-D to end the bar code. If you want to use Access, Excel, or Word to print bar codes, see the section on VBA macro functions in this documentation. So a sample string to print might be a012345b. This becomes the bar code:

a012345b a012345b standard codabar

a012345b a012345b human readable codabar

ASPECT RATIOS

The wide to narrow ratio for these bar code fonts is 3.0. This Codabar font set includes 6 different aspect ratios of each bar code: .25 *(A),* .5 *(B),* .75 *(C),* 1.0 *(D),* 1.5 *(E),* and 2.0 *(F). (See Chart 5.)* 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 about 8.33 characters/inch on 300 dpi printers, or 16.6 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.

EAN BAR CODES

This package includes five EAN bar code fonts in scalable TrueType format, a Windows utility *(BarEAN)* to help you make bar codes, plus VBA macros for Excel, Access, and Word. The EAN bar code may be used in 8 digit *(EAN 8),* 13 digit *(EAN 13)*, or Bookland *(an EAN 13 code with a 5 digit supplemental code)* formats.

EAN 8 bar codes are fairly simple to implement. Both EAN 13 and Bookland *(which is just a variation of EAN13)* are very complicated bar codes and require our utility, or a program of your own, to implement. All EAN bar codes require a checksum *(check digit)*. Each of these EAN bar codes comes in five different aspect ratios, allowing you to print bar codes with the same width but at different heights.

Chart 6

Typeface:	EAN aa	EAN a	EAN b	EAN c	EAN d
EAN 8	[ABCD|4565]				
EAN 13	!BCDEFG|789012]				
Book-land	*HYRFWI|302898] /j.u.f.a.q				

EAN BAR CODES

This EAN bar code font set includes the following versions of the bar code font: EAN aa, EAN a, EAN b, EAN c, and EAN d. Each bar code font can print EAN 8, EAN 13, and Bookland bar codes. EAN bar codes always have a Start character, data, a guard bar, more data, a checksum, and a Stop character. The 5 digit supplemental bar code is a bit more complicated. . If you just want to print a few bar codes in your existing Windows programs, try our BarEAN utility program. *(Skip ahead to the BarEAN section.)* If you want to use Access, Excel, or Word to print bar codes, see the section on our VBA functions.

EAN bar codes represent the numbers 0-9 with three different bar code patterns. Thus, the bar code patterns for a 0 in Tall A are different from the bar code patterns for a 0 in Tall B. Tall C bar code patterns are the inverse images of Tall A patterns. Bar code data is always encoded using two or three different bar code patterns *(Tall A & C for EAN 8, and Tall A, B, & C for EAN 13).* Table 12 identifies which ASCII characters represent the individual numbers 0-9 for each bar code pattern. Our bar code fonts include representations of the Tall A, B, and C bar code patterns, Start codes with leading numbers plus Start codes without a number, and Short A & B patterns for the 5 digit supplemental code.

Table 12

Start		Tall A		Tall B		Tall C		Short A		Short B	
Number	Character Position	Number	Character Position	Number	Character Position	Number	Character Position	Number	Character Position	Number	Character Position
0	!	0	A	0	Q	0	0	0	a	0	q
1	"	1	B	1	R	1	1	1	b	1	r
2	#	2	C	2	S	2	2	2	c	2	s
3	$	3	D	3	T	3	3	3	d	3	t
4	%	4	E	4	U	4	4	4	e	4	u
5	&	5	F	5	V	5	5	5	f	5	v
6	'	6	G	6	W	6	6	6	g	6	w
7	(	7	H	7	X	7	7	7	h	7	x
8	)	8	I	8	Y	8	8	8	i	8	y
9	*	9	J	9	Z	9	9	9	j	9	z
Start	[	Stop	]	Guard	|	Start 5	/	Space 5	.		

EAN 8

EAN 8 bar codes always begin with the *Start code* **[,** have 4 data characters using the Tall A characters *(numbers 0-9 are represented by ASCII characters A-J)*, a Guard Bar **|**, three more data characters and one checksum character using the Tall C characters *(numbers 0-9 are represented by ASCII characters 0-9)*, and a *Stop code* **]**. In Excel or Access use our VBA function, EAN8(). So to make an EAN 8 bar code of the data "0923457", we calculate a checksum digit (4) and we encode:

Data:	09234574
Left/Right	0923 4574
Map characters	AJCD 4574
Plus start, etc	[AJCD | 4574]
Complete string	[AJCD|4574]
Bar Code	

EAN 8 checksums are calculated as follows. Assign the right-most digit an "odd" value, and then alternate even/odd assignments with the rest of the digits. Add all the odd position digits and multiply that by 3. Now add to that the sum of all the even position digits. The checksum is the smallest number that can be added to that sum to make it an even multiple of 10.

```
Data:          0 9 2 3 4 5 7
Position:      O E O E O E O
Odd sum:       0 + 2 + 4 + 7 = 13, 13 * 3 = 39
Even sum:        9 + 3 + 5 = 17
Odd plus even:17 + 39 = 56
Checksum:      4 (56 + 4 = 60, which is an even multiple of 10)
```

EAN 13

EAN 13 bar codes always begin with a special numbered *Start code.* This is just a Start character with a number from 0-9 printed to the left of the start pattern. The number represents the left-most digit of the bar code. *(Numbers 0-9 are represented by the sequential ASCII characters !-*)* **Note** that the Start Character does not actually encode any data. In Excel or Access use our VBA function, EAN13().

The Start code is followed by 6 data characters using a *mixture* of the Tall A & B characters *(numbers 0-9 are represented by ASCII characters A-J and Q-Z respectively)*, a Guard Bar |, 5 more data characters and one checksum character using the Tall C characters *(numbers 0-9 are represented by ASCII characters 0-9)*, and a *Stop code* **]**.

An EAN 13 bar code has room for only 12 digits of bar coding data, but actually represents a 13-digit number. The 13th digit *(the left-most digit)* is encoded by using a pattern of different bar codes *(Tall A & B)* to represent five of the significant digits *(left side digits 11-7)* of the bar code. The right side of the bar code *(digits 6-1)* is encoded using Tall C. The 12th digit is always encoded in Tall A.

To make an EAN 13 bar code of the 12 digits of data "292345789024", we calculate a checksum digit (7) and we encode:

```
Position:          13  12  11  10  9   8   7   6   5   4   3   2   1
Data:              2 9 2 3 4 5 7 8 9 0 2 4 7

Start/Left/Right   2 9 23457   890247
Map characters     # J CTUFX   890247
Plus start, etc    # J CTUFX | 890247 ]
Complete string    #JCTUFX|890247]
```

Bar Code 

Note that the left digit group "23457" does not use just the Tall A or the Tall B characters to represent the number "23457". In this example, the 2 uses Tall A, 3 & 4 use Tall B, 5 uses Tall A, and 7 uses Tall B. The reason for this is that the 13th digit *(a "2")* is encoded in the pattern of switching from the Tall A font to the Tall B font! To select the encoding for these 5 digits use Table 13. Look up the value of the 13th digit of your bar code. That row in the table will tell you whether to use Tall A, or Tall B to encode each of the data digits 11 – 7. In our example, the 13th digit is a 2. From the table we see that Digit 11 should be encoded using Tall A, Digit 10 is Tall B, Digit 9 is Tall B, digit 8 is Tall A, and Digit 7 is Tall B. So Digit 11, *a "2"*, uses Tall A where a "2" is represented by the letter "C". Digit 10, a "3", uses Tall B where a "3" is represented by the letter "T". *(These values come from Table 13.)*

Table 13

Digit 13		Digit 11	Digit 10	Digit 9	Digit 8	Digit 7
0		A	A	A	A	A
1		A	B	A	B	B
2		A	B	B	A	B
3		A	B	B	B	A
4		B	A	A	B	B
5		B	B	A	A	B
6		B	B	B	A	A
7		B	A	B	A	B
8		B	A	B	B	A
9		B	B	A	B	A

EAN 13 checksums are calculated as follows. Assign the right-most digit an "odd" value, and then alternate even/odd assignments with the rest of the digits. Add all the odd position digits and multiply that by 3. Add to that the sum of all the even position digits. The checksum is the smallest number that can be added to that sum to make it an even multiple of 10.

```
Data:          2 9 2 3 4 5 7 8 9 0 2 4
Position:      E O E O E O E O E O E O
Odd sum:         9 + 3 + 5 + 8 + 0 + 4 = 29, 29 * 3 = 87
Even sum:      2 + 2 + 4 + 7 + 9 + 2 = 26
Odd plus even:87 + 26 = 113
Checksum:      7 (113 + 7 = 120, which is an even multiple of 10)
```

BOOKLAND

Bookland bar codes are printed on the back of books. They have two parts: an ISBN number and the book price. Currently, Bookland bar codes always start with the number "978", followed by the left-most 9 digits of the ISBN number. *(The right-most digit of the ISBN number is a checksum and it is not used.)* As of January 1, 2007, all ISBN numbers will be 13 digits long and always start with 978, or when those numbers run out, 979. The main portion of a Bookland bar code is simply an EAN 13 bar code. This includes the "978" or "979" and the 10 digit ISBN number. The pricing information is carried in a 5 digit supplemental code, positioned to the right of the EAN 13 bar code. In Excel or Access use our VBA function, Bookland().

As stated above, a Bookland bar code is actually composed of two separate bar codes. The first bar code is an EAN 13 bar code that represents the book's ISBN number. The second piece of the Bookland bar code is the 5 digit supplemental code. The first digit of the supplemental code indicates the currency the price is stated in. *(The number 5, for example, indicates the book is priced in US dollars.)* The next 4 digits show the price of the book. So a book that costs $12.95 would have a 5 digit supplemental code of "51295". Books priced at $99.99 or higher are encoded as "59999". Finally, if no price is stated for the book, a code of "90000" is used.

To make a Bookland bar code, you first have to convert your ISBN number and your pricing information into the proper format. Take your ISBN number and drop the right-most *(least significant)* digit. Place the numbers "978" before your ISBN number. So if your ISBN number was 1-56830-289-4, you drop the "4", add "978", and you get a number "978156830289". *(If you are using our BarEAN utility, you just enter the ISBN number. The utility figures out the rest.)* You encode this 12 digit number as a standard EAN 13 bar code. *(The checksum is the 13th digit.)* See the previous section on EAN 13 encoding for details.

*HYRFWI|302898] ➔ 9 781568 302898

Next, translate your book price into a 5 digit supplemental code. If your book sells for $9.95 in US dollars, your supplemental number is 50995 *(5 for US dollars, and 0995 for the price).* Finally, you need to calculate a parity pattern for this supplemental number. Assign the right-most digit an "odd" value, and then alternate even/odd assignments with the rest of the digits. Add all the odd position digits and multiply that sum by 3. Add all the even position digits and multiply that by 9. Add those two sums together. The parity pattern is the number in the units position *(or the remainder when you divide by 10).*

```
Data:           5 0 9 9 5
Position:       O E O E O
Odd sum:        5 + 9 + 5 = 19, 19 * 3 = 57
Even sum:         0 + 9 = 9, 9 * 9 = 81
Odd plus even: 57 + 81 = 138
Parity Pattern: 8
```

The five digit supplemental code uses the Short A and Short B bar code patterns *(from Table 12).* You select the bar code pattern *(Short A or B)* for each digit of the supplemental code by looking up the parity pattern you just calculated in Table 13. For the 5 digit supplemental code "50995" we calculated a parity pattern of 8. Looking up the "8" in Table 13, we get a value of "ABAAB". That means the first *(left-most)* digit *(5)* is encoded in Short A, the "0" in Short B, the first "9" in Short A, the next "9" in Short A, and the final "5" in Short B.

Table 14

Number	Pattern
0	B B A A A
1	B A B A A
2	B A A B A
3	B A A A B
4	A B B A A
5	A A B B A
6	A A A B B
7	A B A B A
8	A B A A B
9	A A B A B

The last step is to put these bar code patterns together into a full five digit supplemental code. The supplemental code must be separated from the EAN13 bar code by a single space. It must have its own Start character, which is the "/", and every digit must be separated from its neighbor by a character delineator- the period, ".". For our example data *(50995, parity pattern 8)* this gives us the string:

/f.q.j.j.v ➔ 

To put this all together, add the two strings, placing a space between them. The result is:

*HYRFWI|302898] /f.q.j.j.v ➔ 

ASPECT RATIOS

This complete EAN bar code font set includes five different aspect ratios of each bar code: 0.25 *(aa),* 0.5 *(a),* 0.75 *(b),* 1.0 *(c),* 1.25 *(d). (See Chart 6.)* When you are building a bar code, start with the c version from our bar code fonts. 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 aa, a, b, or d versions. The b version will print a bar code of exactly the same length as the c, but at .75 times the bar height. The d version will print a bar code of exactly the same length as the c, but at 1.25 times the bar height.

OCR FONTS

Chart 7

Typeface Name	*Sample*
OCR-A	ABCDEFabcdef0123
OCR-B	ABCDEFabcdef0123

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.

Chart 8

Typeface Name	*Bar Code*	
Postal	(601832)	Standard 18 point

The Postal Bar Code soft font contains bar code patterns for the numbers 0 - 9, as well as the lead-in and out characters. The checksum is printed using a digit from 0 - 9. To print a Postal Bar Code *(ignoring the checksum calculation 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, you 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). Therefore a checksum of 5 is needed 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 BAR CODES

This package includes five UPC-A and five UPC-E bar code fonts in the scalable TrueType format, a Windows utility, BarUPC, which helps you make bar codes, and VBA macros for Excel, Access, and Word. Each UPC-A bar code encodes an 11 digit number, while UPC-E bar codes encode 6 digit numbers. Both bar code fonts also support optional 2 or 5 digit supplemental codes.

UPC-A bar codes are complicated and you can not simply type numbers in the bar code font. However, they can be calculated by hand. UPC-E codes are more involved and require our utility, our VBA functions, or a program of your own to implement. All UPC bar codes require a checksum *(check digit)*. Each of these UPC bar codes comes in five different aspect ratios, allowing you to print bar codes with the same width but at different heights.

Chart 9

UPCA a	UPCA b	UPCA c	
			
UPCE a	**UPCE b**	**UPCE c**	
[XBGJZQ]6	[XBGJZQ]6	[XBGJZQ]6	

	UPC-A	UPC-A		UPC-E	UPC-E
Digit mental				[XBGJZQ] /t.v	[XBGJZQ] /t.v
Digit mental				[XBGJZQ] /t.f.j.b.r	[XBGJZQ] /t.f.j.b.r

UPC BASICS

This UPC bar code set includes five separate fonts for each of the two UPC bar code types *(UPC-A and UPC-E)*: UPC-A aa, UPC-A a, UPC-A b, UPC-A c, UPC-A d, UPC-E aa, UPC-E a, UPC-E b, UPC-E c, and UPC-E d. *(See Chart 9.)* Each bar code font can print either the UPC-A or the UPC-E bar codes, plus an optional 2 or 5 digit supplemental code. If you just want to print a few bar codes in your existing Windows programs, try our BarUPC utility program. *(Skip ahead to the BarUPC section.)* If you want to use Access, Excel, or Word to print bar codes, see the section on our VBA functions.

	UPC-A		UPC-E	
! HBAAB | 00699 V [XBGJZQ]6				
	start	data guard data	stop	start data stop

UPC-A bar codes always have a Start character, 5 data characters, a guard bar, 5 more data characters, and a checksum / Stop character. UPC-E bar codes always have a Start character, 6 data characters, and a Stop

character. The UPC-E checksum digit is encoded as a parity pattern in that 6 digit number. The 2 or 5 digit supplemental bar codes are a bit more complicated.

UPC-A

UPC-A bar codes represent the numbers 0-9 with two different bar code patterns. Half of the bar code goes to the left of a "guard" bar, and the other half goes to the right of it. The bar code patterns for a 0 on the left side of the bar code are the mirror image of the bar code patterns for a 0 on the right side of the same bar code. In addition, the height of the bar code Start and Stop characters is different from the height of the rest of the data in the bar code. Table 15 identifies which ASCII characters represent the individual numbers 0-9 for each bar code pattern. Our bar code fonts also include representations of Short Odd and Even patterns for the 2 and 5 digit supplemental codes.

Table 15

← UPC-A Codes →										← 2 & 5 digit supplemental →			
Start		**Left / Odd**		**Right / Even**		**Stop**		**Short Odd**		**Short Even**	
Number	**Character Position**	**Number**	**Character Position**	**Number**	**Character Position**	**Number**	**Character Position**	**Number**	**Character Position**	**Number**	**Character Position**
0	!	0	A	0	0	0	Q	0	a	0	q
1	"	1	B	1	1	1	R	1	b	1	r
2	#	2	C	2	2	2	S	2	c	2	s
3	$	3	D	3	3	3	T	3	d	3	t
4	%	4	E	4	4	4	U	4	e	4	u
5	&	5	F	5	5	5	V	5	f	5	v
6	'	6	G	6	6	6	W	6	g	6	w
7	(	7	H	7	7	7	X	7	h	7	x
8	)	8	I	8	8	8	Y	8	i	8	y
9	*	9	J	9	9	9	Z	9	j	9	z
Start	[	Stop	]	Guard	|	Start 5	/	Space 5	.		

UPC-A bar codes always begin with a special numbered *Start Code.* This is a Start character combined with a leading digit from 0-9. A human readable number is printed to the left of the Start pattern. This number represents the left-most digit of the bar code. *(Start numbers 0-9 are represented by the sequential ASCII characters ! through *.)*

The Start Code is followed by 5 data characters using the Left/Odd characters *(numbers 0-9 are represented by ASCII characters A through J)*, a Guard Bar |, 5 more data characters using the Right/Even characters *(numbers 0-9 are represented by ASCII characters 0 through 9),* and a checksum character using the Stop characters *(numbers 0-9 are represented by ASCII characters Q through Z)*.

To make a UPC-A bar code of the 11 digits of data "71456900173", we calculate a checksum digit (5) and we encode:

Position:	12 11 10 9 8 7 6 5 4 3 2 1
Data:	7 1 4 5 6 9 0 0 1 7 3 5
Start/Left/Right/Checksum	7 14569 00173 5
Add guard bar	7 14569 | 00173 5
Map characters	(BEFGJ | 00173 V
Complete string	(BEFGJ|00173V
Bar Code	(BEFGJ|00173V

UPC-A checksums are calculated as follows. Take the 11 digit UPC number and assign the right-most digit an "odd" value. Next alternate even/odd assignments for the rest of the digits. Add all the odd position digits and multiply that sum by 3. Add to that first sum the sum of all the even position digits. The checksum is the smallest number that can be added to that sum to make it an even multiple of 10.

```
Data:          7 1 4 5 6 9 0 0 1 7 3
Position:      O E O E O E O E O E O
Odd sum:               7 + 4 + 6 + 0 + 1 + 3 = 21, 21 * 3 = 63
Even sum:        1 + 5 + 9 + 0 + 7 = 22
Sum odd plus even:    63 + 22 = 85
Checksum:      5 (85 + 5 = 90, which is an even multiple of 10)
```

UPC-E

UPC-E bar codes are an abbreviated form of UPC-A codes. UPC-E bar codes are designed for use in small areas. To create a UPC-E bar code, start with a UPC-A bar code number. Your UPC-A bar code *must* start with a zero, and it *must* have at least 4 zeros in the rest of the bar code data. UPC-E bar codes appear to be 6 characters long, with a checksum character encoded as a parity pattern in the bar code data. Table 16 identifies which ASCII characters represent the individual numbers 0-9 for each bar code pattern. Our bar code fonts also include representations of Short/Odd and Even patterns for the 2 and 5 digit supplemental codes.

Table 16

← UPC-E Codes →						← 2 & 5 digit supplemental →			
Label		**Left / Odd**		**Right / Even**		**Short Odd**		**Short Even**	
Number	**Character Position**	**Number**	**Character Position**	**Number**	**Character Position**	**Number**	**Character Position**	**Number**	**Character Position**
0	0	0	A	0	Q	0	a	0	q
1	1	1	B	1	R	1	b	1	r
2	2	2	C	2	S	2	c	2	s
3	3	3	D	3	T	3	d	3	t
4	4	4	E	4	U	4	e	4	u
5	5	5	F	5	V	5	f	5	v
6	6	6	G	6	W	6	g	6	w
7	7	7	H	7	X	7	h	7	x
8	8	8	I	8	Y	8	i	8	y
9	9	9	J	9	Z	9	j	9	z
Start	[	Stop	]			Start 5	/	Space 5	.

UPC-E bar codes always begin with the *Start Code* **[, and** have 6 data characters using a mixture of the Left/Odd *(numbers 0-9 are represented by ASCII characters A-J)* and Right/Even characters *(numbers 0-9 are represented by ASCII characters Q-Z)*, and a *Stop Code* **]**. To make a UPC-E bar code from the UPC-A data "07110000455", you first calculate a checksum digit (7). Then use the following rules to determine which digits from the UPC-A code go into the UPC-E code. Once you have determined which digits go into the UPC-E code, use the checksum to generate a parity pattern among those digits. See Table 17 for the checksum to parity pattern conversion.

UPC-A code
0 71100 00455 7
0 5 digit manufacturer number 5 digit item number checksum

1) If the 5 digit manufacturer's number ends in 000, 100, or 200, the 5 digit item number may range from 00000 to 00999. The 6 characters in the UPC-E code are: the first 2 characters of the

manufacturer's number, the last 3 characters of the item number, followed by the 3rd character of the manufacturer's number.

2) If the 5 digit manufacturer's number ends in 300, 400, 500, 600, 700, 800, or 900, the 5 digit item number may range from 00000 to 00099. The 6 characters in the UPC-E code are: the first 3 characters of the manufacturer's number, the last 2 characters of the item number, followed by the character "3".
3) If the 5 digit manufacturer's number ends in 10, 20, 30, 40, 50, 60, 70, 80, or 90, the 5 digit item number may range from 00000 to 00009. The 6 characters in the UPC-E code are: the first 4 characters of the manufacturer's number, the last character of the item number, followed by the character "4".
4) If the 5 digit manufacturer's number does not end in zero, the 5 digit item number may range from 00005 to 00009. The 6 characters in the UPC-E code are: all 5 characters of the manufacturer's number and the last character of the item number.

Table 17

Checksum Value	Character Parity vs Position					
	1	2	3	4	5	6
0	E	E	E	O	O	O
1	E	E	O	E	O	O
2	E	E	O	O	E	O
3	E	E	O	O	O	E
4	E	O	E	E	O	O
5	E	O	O	E	E	O
6	E	O	O	O	E	E
7	E	O	E	O	E	O
8	E	O	E	O	O	E
9	E	O	O	E	O	E

UPC-E Example 1

Data:	07110000455
Manufacturer number	71100
Item number	00455

Since the last 3 characters of the manufacturer's number are 100, the 6 digits of the UPC-E code are:

First 2 characters of manufacturer's number	71	
Last 3 characters of the item number	455	
Third character of manufacturer's number	1	
Checksum (from UPC-A number)	7	(see UPC-A for example)
Bar Code data	714551 7	
Parity pattern for digits, from Table 17	EOEOEO	(look up parity pattern for checksum of 7)
Map characters using even/odd parity	XBUFVB	
Add start, stop, and checksum display	[XBUFVB]7	
Bar Code	[XBUFVB]7	

UPC-E Example 2

Data:	09135700009
Manufacturer number	91357
Item number	00009

Since the last 3 characters of the manufacturer's number are 100, the 6 digits of the UPC-E code are:

First 5 characters of manufacturer's number	91357
Last character of the item number	9
Checksum (from UPC-A number)	6 (see UPC-A for example)
Bar Code data	913579 6
Parity pattern for digits, from Table 17	EOOOEE (look up parity pattern for checksum of 6)
Map characters using even/odd parity	ZBDFXZ
Add start, stop, and checksum display	[XBUFVB]6
Bar Code	[ZBDFXZ]6

5 DIGIT SUPPLEMENTAL CODES

Both UPC-A and UPC-E bar codes support the same 2 and 5 digit supplemental codes. These supplemental codes let you add a two or a five digit number to any UPC bar code. All supplemental calculations are the same for both UPC-A and UPC-E bar codes. First, take your 5 digit supplemental number and calculate a parity pattern for it. Assign the right-most digit an "odd" value, and then alternate even/odd assignments with the rest of the digits. Add all the odd position digits and multiply that by 3. Add all the even position digits and multiply that by 9. Add those two sums together. The parity pattern is the number in the ones position *(the remainder when you divide by 10).*

```
Data:          7 1 3 0 5
Position:      O E O E O
Odd sum:       7 + 3 + 5 = 15, 15 * 3 = 45
Even sum:        1 + 0 = 1, 1 * 9 = 9
Odd plus even:  45 + 9 = 54
Parity Pattern: 4
```

The five digit supplemental code uses the Short/Odd and Short/Even bar code patterns *(from Table 16 or 17).* You select the bar code pattern *(Short/Odd or Short/Even)* for each digit of the supplemental code by looking up the parity pattern you just calculated in Table 17. For the 5 digit supplemental code "71305" we calculated a parity pattern of 4. Looking up the "4" in Table 17, we get a value of "OEEOO". That means the first *(left-most)* digit *(7)* is encoded in Short/Odd, the "1" in Short/Even, the first "3" in Short/Even, the next "0" in Short/Odd, and the final "5" in Short/Odd.

Table 18

Number	Pattern	
0	E E O O O	
1	E O E O O	
2	E O O E O	
3	E O O O E	
4	O E E O O	←
5	O O E E O	
6	O O O E E	
7	O E O E O	
8	O E O O E	
9	O O E O E	

The last step is to put these bar code patterns together into a full five digit supplemental code. The 5 digit supplemental code must be separated from the UPC by a single space. It must have its own Start character, which is the "/", and every digit must be separated from its neighbor by a character delineator, the period, ".". The actual ASCII character you use to represent the number is chosen from the Short/Odd or Short/Even section of the font,

based on the parity pattern chosen by the checksum. For our example data *(71305with parity pattern 4)* this gives the string:

Bar Code data	`71305 4`	
Parity pattern for digits, from Table 18	`OEEOO`	(look up parity pattern for checksum of 4)
Map characters using even/odd parity	`hrtaf`	
Add start, delineators, and stop	`/h.r.t.a.f`	
Bar Code	`/h.r.t.a.f`	

To add this 5 digit code to any UPC code, add the string shown above to the end of your UPC bar code, placing a space between them.

2 DIGIT SUPPLEMENTAL CODES

All supplemental calculations are the same for both UPC-A and UPC-E bar codes. First, take your 2 digit supplemental number *(00 – 99)* and calculate a parity pattern for it. To do this, divide the two digit supplemental number by 4 and use the remainder to look up the parity pattern in Table 19.

Table 19

Remainder	Pattern
0	O O
1	O E
2	E O
3	E E

The final step is to put these bar code patterns together into a full two digit supplemental code. The 2 digit supplemental code must be separated from the UPC code by a single space. It must have its own Start character, which is the "/", and the first and second digits must be separated from each other by a character delineator, the period, ".". The actual ASCII character you use to represent the number is chosen from the Short/Odd or Short/Even section of the font, based on the parity pattern calculated from your division remainder. Suppose we wish to encode 73 as a two digit supplemental code. Then:

Bar Code data	`73`	
Parity pattern for digits, from Table 19	`OE`	(look up parity pattern for remainder of 1)
Map characters using even/odd parity	`ht`	
Add start, delineators, and stop	`/h.t`	
Bar Code	`/h.t`	

To add this 2 digit code to any UPC code, add the string shown above to the end of your UPC bar code, placing a space between them.

BARPLUS UTILITY PROGRAM

The BarPlus utility program has automatically been installed in Windows. This program will demonstrate how to use our bar code fonts and calculate checksums for those bar codes that need them. You can also use the program to print scanable samples of every bar code you enter into the program. To start the program, click on Start, Programs, Elfring Bar Codes Plus, and then Utility program.

To use the BarPlus program, you must first select the type of bar code you want to create at the top of the screen. You have the choice of bar codes: 2/5 interleaved, 3/9, 93, 128, Codabar, EAN, PostNet, or UPC. Make this selection at the top of the program screen. *(Bookland bar codes are listed under the EAN option.)*

Once you have selected a type of bar code, matching options will appear on the program window. You can always choose a font size, except with the PostNet bar code. For most bar codes you can also choose between human readable and standard bar codes. Select the options you want before proceeding.

Next, enter the data you want to convert to a bar code in the data entry window. Once you have entered your text, click on the "**Make a bar code**" 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 to clip board**" button. After this data is stored in the clipboard, you can paste it into any Windows application. *(You will have to change the font back to the matching bar code font after pasting it.)* The final button, **"Print a bar code"** 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 BarPlus program will display an error message**.** Also **note** that if you change any of the bar code symbologies, you must click the "**Make a Bar Code**" button before printing or copying the bar code data.

VBA 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
Bar25I(Text)	Converts the input text 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 25i b, Bar 25i b HR, Bar 25i c, Bar 25i c HR, Bar 25i d, Bar 25i d HR, Bar 25i e, Bar 25i e HR, Bar 25i f, Bar 25i f HR.
Bar25Ics(Text)	Converts the input text 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 25i b, Bar 25i b HR, Bar 25i c, Bar 25i c HR, Bar 25i d, Bar 25i d HR, Bar 25i e, Bar 25i e HR, Bar 25i f, Bar 25i f HR.

Function	Details
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 a, Bar Code 39 a HR, Bar Code 39 b, Bar Code 39 b HR, Bar Code 39 c, Bar Code 39 c HR, Bar Code 39 d, Bar Code 39 d HR, Bar Code 39 e, Bar Code 39 e HR, Bar Code 39 f, Bar Code 39 f HR.
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 a, Bar Code 39 a HR, Bar Code 39 b, Bar Code 39 b HR, Bar Code 39 c, Bar Code 39 c HR, Bar Code 39 d, Bar Code 39 d HR, Bar Code 39 e, Bar Code 39 e HR, Bar Code 39 f, Bar Code 39 f HR.
Bar39fa(Text)	Converts the input text data into a bar code 3/9 in Full ASCII mode. Note your bar code reader must have Full ASCII mode enabled to use this function! The function adds the Start code, converts characters that aren't in standard bar code 3/9 into their Full ASCII character pairs, 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 a, Bar Code 39 a HR, Bar Code 39 b, Bar Code 39 b HR, Bar Code 39 c, Bar Code 39 c HR, Bar Code 39 d, Bar Code 39 d HR, Bar Code 39 e, Bar Code 39 e HR, Bar Code 39 f, Bar Code 39 f HR.

Function	Details
Bar93(Text)	Converts the input text string into bar code 93 format. The function adds the Start code, appends the data, calculates and adds the two checksums, and puts the Stop code at the end. This result must be formatted with one of the following typefaces: Code 93 a, Code 93 a HR, Code 93 b, Code 93 b HR, Code 93 c, Code 93 c HR, Code 93 d, Code 93 d HR, Code 93 e, Code 93 e HR, Code 93 f, Code 93 f HR. Non-legal characters are ignored.
Bar93fa(Text)	Converts the input text string into bar code 93 format, using Full ASCII mode. The function adds the Start code, appends the data, calculates and adds the two checksums, and puts the Stop code at the end. Characters not part of the base 47 character set are converted into character pairs. This result must be formatted with one of the following typefaces: Code 93 a, Code 93 a HR, Code 93 b, Code 93 b HR, Code 93 c, Code 93 c HR, Code 93 d, Code 93 d HR, Code 93 e, Code 93 e HR, Code 93 f, Code 93 f HR.

Function	Details
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 following typefaces: Code 128AB, Code 128AB HR, Code 128AB Short, Code 128AB Short HR, Code 128AB Tall, Code 128AB Tall HR.
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 following typefaces: Code 128AB, Code 128AB HR, Code 128AB Short, Code 128AB Short HR, Code 128AB Tall, Code 128AB Tall HR.
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 following typefaces: Code 128C, Code 128C HR, Code 128C Short, Code 128C Short HR, Code 128C Tall, Code 128C Tall HR.
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 following typefaces: Code 128AB, Code 128AB HR, Code 128AB Short, Code 128AB Short HR, Code 128AB Tall, Code 128AB Tall HR.
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 following typefaces: Code 128AB, Code 128AB HR, Code 128AB Short, Code 128AB Short HR, Code 128AB Tall, Code 128AB Tall HR.
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 the checksum, and puts the Stop code at the end. This result must be formatted with one of the following typefaces: Code 128C, Code 128C HR, Code 128C Short, Code 128C Short HR, Code 128C Tall, Code 128C Tall HR.

Function	**Details**
Codabar(Text)	Converts the input text data into a complete Codabar string. The function verifies the Start code, throws away all characters that aren't in standard Codabar, appends the data, and verifies the Stop code at the end. This result must be formatted with one of the following typefaces: Codabar a, Codabar a HR, Codabar b, Codabar b HR, Codabar c, Codabar c HR, Codabar d, Codabar d HR, Codabar e, Codabar e HR, Codabar f, Codabar f HR.
Codabarss(Text, Start, Stop)	Converts the input text data into a complete Codabar string. The function verifies and then adds the passed the Start code, throws away all characters that aren't in standard Codabar, appends the data, and verifies and adds the passed Stop code at the end. This result must be formatted with one of the following typefaces: Codabar a, Codabar a HR, Codabar b, Codabar b HR, Codabar c, Codabar c HR, Codabar d, Codabar d HR, Codabar e, Codabar e HR, Codabar f, Codabar f HR.

Function	**Details**
EAN8(Number)	Converts the input 7 digit number into a complete EAN 8 bar code. 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 following typefaces: EAN aa, EAN a, EAN b, EAN c, EAN d. Non-numeric data is ignored. If less than 7 digits are supplied, all are replaced with zeros. Any digits past the seventh place are ignored.
EAN13(Number)	Converts the input 12 digit number into a complete EAN 13 bar code. The function adds the Start code, appends the data and guard bar, calculates and adds the checksum, and puts the Stop code at the end. This result must be formatted with one of the following typefaces: EAN aa, EAN a, EAN b, EAN c, EAN d. Non-numeric data is ignored. If less than 12 digits are supplied, all are replaced with zeros. Any digits past the twelfth place are ignored.
Bookland(ISBN,Supplem)	Converts the input ISBN number into a complete Bookland bar code. The function converts the ISBN number into EAN 13 format and adds the 5 digit supplemental code to the end. This result must be formatted with one of the following typefaces: EAN aa, EAN a, EAN b, EAN c, EAN d. Non-numeric data is ignored. If the ISBN number is less than 9 digits it will be replaced with zeros. Any digits past the ninth place are ignored. The supplemental code must be 5 digits long.

Function	**Details**
BarUPCa(Text)	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: UPCA aa, UPCA a, UPCA b, UPCA c, UPCA d.
BarUPCe(Text)	Converts the input 11 or 12 digit number into a complete UPC-E bar code compatible with our UPC-E fonts. UPC-E numbers must Start with a zero, and have limits on the manufacturer's number, and on the product number. See BarUPC.rtf for details on these limits. This function translates the 11 or 12 digit number into the abbreviated UPC-E format. 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: UPCE aa, UPCE a, UPCE b, UPCE c, UPCE d.

Warning!

Unlike our utility program, *BarPlus*, these VBA 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 *BarPlus* utility program to verify that your data is correct before using these VBA functions to mass produce bar codes.

Using VBA 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 Plus package *(probably BarPlus)*. The VBA file **VBbarPlus.bas** should appear there. Select this file and open it. This will add a new module, EBARPLUS, to your spread sheet. This module adds the two functions *(see Table 4)* 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, =Bar93(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 the documentation for applicable font names.

Using VBA 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 Plus package *(probably BarPlus)*. The VBA file **VBbarPlus.bas** should appear there. Select this file and open it. This will add a new module, EBARPLUS, to your database. This module adds these functions to your database and is saved along with 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 =Bar93([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 the documentation for a complete listing of font names.

Using VBA functions in Word

Older versions of Word do not have the ability to directly use VBA functions. *(Word 2000 and newer versions do.)* But you can 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.

You would 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 typeface for that bar code type for that mail merge field. See the chart above for applicable 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 height to width.

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

Checksum: a number calculated from data which is used 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.

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.

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

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

4N899 W Mary Dr, St Charles, IL 60175
Phone: 1-630-440-2456

www.barcodingfonts.com