Contents

Introduction

Barcodes
Barcode fonts
Installing your barcode font
Getting started
dFont
Automation

Barcode types

Code 39
Code 93
EAN and UPC
Code B
Code 11
Codabar
Interleaved 2-of-5
MSI/Plessey
Royal Mail RM4SCC
PostNet
Code 128/ EAN-128
Telepen

Barcodes

Barcodes are patterns of light and dark which may be scanned by an optical scanner and decoded into characters. A complete barcode is referred to as a barcode Symbol and the characters represented by the symbol is called the Code.

Most barcode symbols are made up of a series of parallel, adjacent bars and spaces, where the bars present a dark image and the spaces a light, reflecting image. In most circumstances a human-readable form of the code is printed directly under the barcode symbol - to allow for manual deciphering in case the symbol has been damaged.

Barcodes fall into two main categories - discrete and continuous. Discrete barcodes are those for which each character in the code translates into a character in the symbol, and each character in the symbol is separated from its neighbour by a gap containing no information. A continuous code has no inter-character gaps - each character starts with a bar and ends with a space, and the start of the following character's first bar is immediately after that space. While continuous codes are more space efficient (i.e. no gaps), they do impose far greater tolerance requirements on barcode printing fonts and the printing process.

Some barcodes types support only numbers; others support uppercase letters and numbers, and some support the entire 128 characters of the ASCII character set (i.e. Characters 0 - 127).

Most barcode type employ a start character the beginning of the symbol and a stop character at the end. In general these start and stop characters are not are not reproduced in human readable form under the barcode symbol. For example, in the symbol above the start and stop characters are * characters, but these do not appear under the symbol.

Most barcode types also employ one or more Check digits, which are used by the scanning equipment to ensure that the code has been read correctly. In some cases these check digits are optional, in others mandatory (i.e. the symbol will not scan if the check digit is omitted). Where present, check digits are usually the last characters in the code, although they are not always shown in human readable form under the symbol.

A "self-checking" barcode symbology is one in which a printing defect cannot cause an incorrect character to be substituted for a misread character.

Barcode fonts

Barcode fonts enable you to print barcodes on Windows supported graphics printers. However, the user needs to be aware of a number of factors which determine whether printed barcodes can actually be scanned correctly.

1. The thickness of bars and spaces in barcodes is important. Some types of barcode use only two thicknesses of bar, others use three thicknesses, and others more. Even when you print a barcode using a dLSoft barcode font, you need to ensure that the barcode has not been printed too small - so that within the resolution of the printer a single thickness bar has been printed at the same size as a double thickness bar. Consequently it is essential that you check that a printed barcode is readable using an appropriate scanner or reader.

Barcodes printed by laser printer will, in general, be printed correctly, but codes printed by matrix printers must be reproduced at a large enough scale that the barcodes unit size is at least as large as the printer's pins.

2. Bar thickness reduction: All dLSoft barcode fonts are supplied in three bar thicknesses. The Wide font (and its variants - names ending in W) should be suitable for most 600 and 1200 dpi laser printers - it has the bar/space ratio defined at its correct value. The Regular font (names ending in R) has all bars reduced by 8% and will probably be a better choice for 300 dpi laser printers and good quality ink-jets. The Narrow font (names ending in N) has all bars reduced by 16% and is supplied for users who will be creating master copy which will subsequently be printed using a wet ink technique (in which the ink spreads, so making each bar thicker than in the master). The narrow fonts should only be used if you know that a bar thickness reduction is required. Picking the wrong font usually produces unreadable images! If greater control of bar thickness is needed then an image creating system, such as dBarcode, will be required.

3. Many barcode types may use codes only of a specific length. (e.g. EAN13 requires 13 digits in the code). Some barcode type use specific digits of the code as a checksum - so not every combination of digits can form a legal barcode. dLSoft barcode fonts display as barcode characters the characters you specify. If your barcode type requires start and stop characters and a check digit character you must provide these character within the string of characters you wish to print as a barcode symbol. Furthermore most coding schemes are limited to 32 characters or less.

3. The barcode types support by dLSoft barcode fonts are described in the remainder of this document, along with details of the Start/Stop character and check digit calculations.

4. Users should be aware that it is possible to print barcodes of a specific type and find that normal retail scanners are unable to decode the images. This does not necessarily mean that there is anything wrong with the barcode symbol. Most scanners aimed at the retail market are not programmed to interpret barcode codes reserved for other (e.g. military) use.
5. Space characters are treated as a special case in a number of Windows applications - the font character being ignored and a gap being placed where the space character was expected. In some applications the same behaviour is seen with the non-break space character (ASCII 160). dLSoft barcode fonts which contain a space character reproduce this character at ASCII 159, and this should be used whenever space characters give rise to a gap in barcodes. Even where the space character is not deliberately encoded it may appear as part of a check digit sequence.

Installing your barcode font

To install your font place the disk containing the font in your disk drive. Start the Windows Control Panel and double click on the Fonts icon. The choose Add New Font or Install New Font and select the disk drive containing your font disk.

A list of the fonts on the disk will be displayed and you can select the ones you wish to install. Unless you specifically require the PostScript fonts (e.g. For Adobe Type Manager or downloading to a PostScript printer) select only the TrueType fonts for installation. Also, unless you know that you will be requiring your barcodes to be wet-ink printed, you will probably not need to install the Narrow fonts. If you are using a modern laser printer then you may need only the Wide fonts (those ending in W), while if your printer has less than 600 dpi resolution, or if your printer makes the bars of your barcodes too thick, you will need to install the Regular fonts (those ending in R).

Many font types are supplied in three different height/width ratios - to give you some choice over the aspect ratio of the printed barcode. Because many Windows programs were created when the choice of fonts styles was relatively limited (and may not be able to handle alternative names) we have used the common style names "Normal", "Bold" and "Italic" to represent these styles.

The Normal style gives a height to width ratio which should be acceptable for most purposes. The Bold style reduces the height of the barcode symbol while keeping the width the same as with the Normal style. The Italic style reduces the width of the symbol while keeping the height that specified by the chosen point size.

Note that the Bold and Italic styles cannot be used at the same time - even if your Windows application permits the selection of such attributes.

Some fonts, such as PostNet and RM4SCC, will not scan if the aspect ratio is not correct. These fonts do not have Bold or Italic components.

Getting started

Once you fonts are installed you may like to try to print a barcode. Before you can print a barcode that has a significant chance of being scanned successfully you will probably need to read the part of this document which refers to the barcode type you need. However, for the moment lets just print a single barcode.

Start a suitable Windows application; anything capable of printing and displaying WYSIWYG will do, if in doubt try the Write or WordPad word processor which comes with Windows.

Into a new (or spare) document type 1234 then press return.

Select the text you have typed, then change its font to the dLSoft barcode font listed in your fonts list, and its size to something like 36 point.

You should see a barcode image, and you should be able to print it.

Unfortunately you will probably not be able to scan the printed barcode without inserting the relevant start and stop characters, and in some cases you will also require a check digit. The characters required for these may be determined by running the dFont program. The source code for this program is included for users who wish to use sections of the code to automate this procedure in programmable applications, such as Microsoft Office, Visual FoxPro, Borland Paradox, etc.

For some of our fonts the ` (grave symbol - top left on most UK keyboards) will work as the start character and the ¬ (shifted grave) will work as the stop character. For other the open bracket (will act as the start character and) will act as the stop character. For all of our fonts (except EAN/UPC) the section symbol § (Chr$(167)) will act as the start character and ¬ (Chr$(172)) will act as the stop character [For fonts which encode all the characters on the keyboard (eg. Code 93 and the 128s) the special characters such as start and stop may be entered by holding down the <Alt> key and type 0xyz using the numeric keypad, where xyz is the value of the character. When these special characters are being created by programming, then use Chr$(xyz) in Basic or the equivalent in other languages.]

Position the cursor before the beginning of the barcode image on screen and type the start character. DO NOT press the enter key. Then position the cursor just after the end of the barcode and type the stop character.

You barcode will be a little longer now, but when its printed this time it should scan - even if the scanner complains that the check digit is wrong. To get that right you will need to start reading the section on your barcode type.

dFont

The dFont program enables you to see which characters need to be used to create a barcode representing a particular Code, including any start, stop and check characters.

To use dFont follow this simple procedure:

1. From the drop down list of barcode types select the code type you require.
2. If you require the normal check digits in your symbol, check the "Include check digits" checkbox. [Note: for code type which always require mandatory check digits the check digits will be included whether you check this box or not.]
3. Type the code you require into the box labelled "Enter Code".
4. Push the recalculate button. The string you need to create the symbol with your dLSoft barcode font will appear in the box labelled "String required".
5. If you wish you can copy the string required to the clipboard for pasting into another Windows application.

For characters which are not normally visible on screen, one of the alternative (visible) characters used in the dLSoft barcode font will be displayed along with the xyz value which may be used to produce that character - either using Chr$(xyz) in Basic or by holding down the Alt key and type 0xyz on the numeric keypad.

Automation

If you are planning to use many barcodes within programmable spreadsheet or database applications, then you will probably wish to automate the process of inserting the start and stop codes and the check digits.

This should prove relatively straightforward because most applications have a programming language. You can copy sections of the dFont helper program supplied with your font into the macro or module section of your application.

The basic principle is to take the string you wish to convert into a barcode (STRING1$) and add to it the start character, check digit (if required) and stop character, to create a second string (STRING2$) which then appears on forms or reports and is set to display in your barcode font at a suitable size. In most cases the font characteristics are set permanently (e.g. within a properties window), while the creation of the string to display is handled by a macro or module.

So your module may contain code such as

STRING2$=start$ + STRING1$ + check$ + stop$

and STRING2$ becomes the content of the target field or cell.

Code 39

Code 39 is a discrete "self-checking" code (i.e. it has inter-character gaps) and is undoubtedly the most commonly used code outside retail labelling.

The standard Code 39 symbology supports upper case letters, numbers and the following additional characters; SPACE, HYPHEN/MINUS (-), POINT(.), PLUS(+), DIVIDE(/), DOLLAR($) and PERCENT(%). No other characters are allowed!

The Extended Code 39 symbology supports the full ASCII character set - although it does this at a price by using two barcode characters to represent each code character (the a is represented in the barcode by +A). Extended Code 39 symbols can thus become rather long, creating difficulties for the scanner.

This code must begin with a start character and must end with a stop character. The start character is § (Chr$(167)) and the stop character is ¬ (Chr$(172)).

For convenience in typing in Code 39 [but NOT in Extended Code 39] the open bracket (may be used as the start character, and the close bracket) may be used as the stop character.

Code 39 symbols are self-checking so are normally used without a check digit. However, they may employ an optional check digit as the last character before the stop character.

The check digit is calculated using a modulo 43 algorithm, using the value of each character as follows:

character	**value**
0 - 9	0-9
A-Z	10 - 35
-	36
.	37
(space)	38
$	39
/	40
+	41
%	42
*	43 (i.e. ignored)

The check digit is calculated by summing the character values to give Sum, and using the formula

checkdigitvalue = Sum Mod 43

The check digit is then the corresponding character in the table above.

The dFont program provided with the dLSoft Barcode fonts calculates check digits and for both Code 39 and Extended Code 39, and the source code is provided and may be adapted to the users needs - for example, for automating the calculation in a spreadsheet or database.

Note that Code39 is supplied as a different font from Extended Code 39 - even though the upper case letters and numbers are the same in both fonts. This is because some character are not the same. For example with + in Code 39 is represented as /K in Extended Code 39 - to avoid the sequence +A being misinterpreted.

For convenience of handling characters with ASCII values of less than 32 are reproduced in the font at the character positions chr$(192) - chr$(223).

Code 93

Code 93 is a continuous symbology and is much more compact than Code 39 to which it is closely related.

The standard Code 93 symbology supports upper case letters, numbers and the following additional characters; SPACE, HYPHEN/MINUS (-), POINT(.), PLUS(+), DIVIDE(/), DOLLAR($) and PERCENT(%). No other characters are allowed!

The Extended Code 93 symbology supports the full ASCII character set - although it does this at a price by using two barcode characters to represent each code character (the a is represented in the barcode by a special character followed by A).

Code 93 symbols are generally contain two check digits calculated using a modulo 47 algorithm, using the value of each character as follows:

character	**value**
0 - 9	0-9
A-Z	10 - 35
-	36
.	37
(space)	38
$	39
/	40
+	41
%	42
shift 1 Chr$(168) ¨	43
shift 2 Chr$(169) ©	44
shift 3 Chr$(170) ª	45
shift 4 Chr$(171) «	46

The check digits are calculated by summing the weighted character values to give sums t1 and t2,

```
j = (n - (i - 1)) Mod 21        ' check digits weighted sums into t1 and t2
t1 = t1 + j * chn
j = ((n - i + 2) Mod 16)
t2 = t2 + j * chn
```

where n is the number of characters in the string (not including the check digits), i is the position in the string (starting at 1 on the left), and chn is the value of the ith character taken from the table above.

Once the sums have been calculated the modulo 47 remainders are the values of the two check digits, the character for which are given in the table above

```
t1 = (t1 Mod 47)                    ' modulo 47 checksum
t2 = ((t2 + t1) Mod 47)
```

The two check digit characters must be appended to the code, which becomes

§code(t1)(t2)¬

This code must begin with a start character and must end with a stop character. The start character is § (Chr$(167)) and the stop character is ¬ (Chr$(172)).

The dFont program provided with the dLSoft Barcode fonts calculates check digits and for both Code 93 and Extended Code 93, and the source code is provided and may be adapted to the users needs - for example, for automating the calculation in a spreadsheet or database.

For convenience of handling characters with ASCII values of less than 32 are reproduced in the font at the character positions chr$(192) - chr$(223).

EAN and UPC

The EAN and UPC codes are the most common retail codes and are to be found on virtually every item in a supermarket.

The standard EAN-13 code contains 13 digits and is a superset of the UPC-A code (which contains 12 digits). Confusingly the 13th digit of the EAN code is actually the leftmost digit, and this is not encoded in bars and spaces, but in the combination of coding rules used to encode the next six digits. The UPC coding of the first six digits uses only one of the coding rules of the EAN standard. As a result, EAN and UPC codes are very similar and can be read by the same scanners.

One of the problems with this approach is that there is not a one-to-one correspondence between a digit and a pattern of bars and spaces; for example, the digit 0 may be encoded in three different patterns in an EAN symbol. The three different patterns are called Sets A, B and C. Consequently three different characters are required to represent 0 within a barcode font - the dLSoft EAN/UPC font uses **A**, **a** and **0**, for set A, B and C respectively as shown in the Character sets Table below.

One of the cosmetic features of EAN and UPC symbols is that the start and stop characters, and the "centre marker" character (which merely separates the left hand six characters from the right hand six) are usually printed slightly longer than the encoded characters. The elongation of these bars has no effect on the machine readability of the symbol.

The dLSoft EAN/UPC font provides two sets of characters, one which contains just the bars and spaces and in which the start, stop and centre marker characters are the same height as the coded characters, and one set in which the barcode characters have their numeric values reproduced underneath and in which the start, stop and centre marker are elongated.

In both cases the normal start and stop characters are [and] respectively and the centre marker is /. For the special case of UPC-E codes the start and stop characters are { and } respectively. UPC-E does not use a centre marker.

EAN and UPC codes both require a check digit, the value of which may be obtained as follows:

Starting with the rightmost digit (excluding the check digit), find the sum of the alternate digits; multiply this sum by 3 and add to the summ of the remaining digits. The check digit is the number which when added to this result gives a multiple of 10.

The right-hand six characters in both EAN and UPC symbols are encoded using characters from Set C. The left hand six digits in UPC codes are encoded using Set A. The left-hand six digits (excluing the 13th) in EAN codes are encoded using a combination of Sets A and B - and the combination of sets is used to encode the 13th digit as follows:

Digit	**n1**	**n2**	**n3**	**n4**	**n5**	**n6**
0	A	A	A	A	A	A
1	A	A	B	A	B	B
2	A	A	B	B	A	B
3	A	A	B	B	B	A
4	A	B	A	A	B	B
5	A	B	B	A	A	B
6	A	B	B	B	A	A
7	A	B	B	B	A	A
8	A	B	A	B	B	A

9 A B B A B A

The dFont program provided with the dLSoft Barcode fonts calculates check digits and encoding patterns for both EAN and UPC-A codes, and their minimal size relatives (EAN-8 and UPC-E), and the source code is provided and may be adapted to the users needs - for example, for automating the calculation in a spreadsheet or database.

Users wishing to use their own encoding schemes should note that the UPC-E stop character is NOT the same as UPC-A or EAN stop characters (because it needs to indicate the direction of scanning). The UPC-E stop character is provided in the dLSoft EAN/UPC fonts as }.

More:

Supplementary codes

Light margin indicators

EAN/UPC Character Sets Table

Supplementary codes

EAN and UPC-A codes may include 2 digit or 5 digit supplementary codes (Add-on codes) which may be encoded using character sets A and B. However, these are normally reproduced with their number values above the bars - rather than underneath the bars as in the case of the main code. In order to accommodate this feature, character sets A and B are also provided in the dLSoft barcode font as character sets ExtA and ExtB as shown in the Character sets Table below.

2-digit supplementaries do not use a check digit for the supplementary, but the encoding of the 2 digits does depend on the value of the add-on. The character sets for the two digits depends on the remainder of dividing the value of the add-on by 4:

Remainder	lefthand digit	righthand digit
0	A	A
1	A	B
2	B	A
3	B	B

5-digit supplementaries do use a check digit calculated by taking three times the sum of digits 1, 3 and 5, plus nine times the sum of digits 2 and 4, and using the units value of the result. Encoding of the 5 digits is then determined by the value of the check digit as follows:

Check digit	n1	n2	n3	n4	n5
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	A	A	A

This table (and the chek value calculation) differs from that used for the main EAN/UPC barcode symbol.

Light margin indicators

EAN barcodes are normally prefixed by the human-readable character form of the 13th digit (which is actually the leftmost digit) and this does not appear under the bars, but in the light margin before the barcode symbol - where it indicates the region in which no other printing should appear. The dLSoft barcode font includes the numerals 0-9 without bars in the ASCII characters 33-41 (the Prefix character set, see below), so that the 13th digit may be printed in the light margin. The righthand light margin is indicated by a > symbol.

The Prefix character set may also be used to print the leading 0 and the check digit for UPC-A codes. These are normally printed at a smaller size than the other human readable characters, so are provided in a UPC Prefix character set (ASCII 161-170). To obtain the correct spacing, the check digit human readable character should be preceded by a space.

The light margin indicators for EAN-8 codes are < on the left and > on the right, and the light margin indicator for EAN supplementary codes is > at the level of the supplementary characters, and provided in the font as . (dot). To obtain the correct margin position this should be preceded by a space.

EAN/UPC Character Sets Table

Code	Formatted	Unformatted (ASCII)
Start	[	219
Stop	]	221
Centre Marker	/	175
Add-on start	+	171
Add-on inter-character	-	173
EAN Prefix	ASCII 33-42	
Margin Indicators	< > and .	
UPC-E start	{	251
UPCE-Stop	}	253
UPC Prefix/Check	ASCII 161-170	

Fully formatted Codes

Digit	Set A	Set B	Set C	ExtA	ExtB	Prefix
0	A	a	0	M	m	!
1	B	b	1	N	n	"
2	C	c	2	O	o	#
3	D	d	3	P	p	$
4	E	e	4	Q	q	%
5	F	f	5	R	r	&
6	G	g	6	S	s	'
7	H	h	7	T	t	(
8	I	I	8	U	u	)
9	J	j	9	V	v	*

Unformatted Codes (ASCII values)

Digit	Set A	Set B	Set C	ExtA	ExtB
0	193	225	176	205	237
1	194	226	177	206	238
2	195	227	178	207	239
3	196	228	179	208	240
4	197	229	180	209	241
5	198	230	181	210	242
6	199	231	182	211	243
7	200	232	183	212	244
8	201	233	184	213	245
9	202	234	185	214	246

Code B

Code B is a numeric only code and does not use a check digit, so there is no Code B entry in the dFont program.

This code must begin with a start character and must end with a stop character. The start character is § (Chr$(167)) and the stop character is ¬ (Chr$(172)).

For convenience in typing the open bracket (may be used as the start character, and the close bracket) may be used as the stop character.

Code 11

Code 11 encodes only the number and a minus sign.

This code must begin with a start character and must end with a stop character. The start character is § (Chr$(167)) and the stop character is ¬ (Chr$(172)).

For convenience in typing the open bracket (may be used as the start character, and the close bracket) may be used as the stop character.

Code 11 provides for a single check digit when the code length is less than 12 characters, and two check digits when the code length is 12 characters or more.

In each case the check digit is calculated using a weighted modulo 11 algorithm. For the first check digit the integer value of each character is multiplied by it the rightmost digit of the character's position (i.e. ranging from 1 to 9 and then restarting) in the code (starting from the right) and with the - sign given the value 10. The sum of these values is calculated and then divided by 11 and the remainder is used as the check digit.

```
' xx$ is the code    First check digit calculation
chn1=0:   chd$=""
n= len(xx$)
For i = n To 1 Step -1
    y$ = Mid$(xx$, i, 1)
    If y$ = "-" Then z = 10 Else z = Asc(y$) - 48'    value of character
    chn1 = chn1 + j * z
    j = j + 1
    If j > 10 Then j = 1
    Next I
t = chn1 Mod 11
If t < 10 Then
    chd$ = chd$ + Chr$(t + 48)' convert to ASCII character 0-9
    Else
    chd$ = chd$ + "-"
    End If
```

The second check digit is calculated in the same way except that the character position weightings range from 1 to 8 and then restart. The first check digit character IS included as the rightmost character in the calculation for the second check digit.

The dFont program provided with the dLSoft Barcode fonts calculates check digits for Code 11 codes, and the source code is provided and may be adapted to the users needs - for example, for automating the calculation in a spreadsheet or database.

Codabar

Codabar is a self-checking, discrete symbology which encodes the numbers and the $: / . + - characters. Codabar is unusual in that the barcode characters have differing widths. Also the symbology allows for several combinations of start and stop character.

Start and stop characters are usually A for start and C for stop,

This code must begin with a start character and must end with a stop character. Start and stop characters are usually A for start and C for stop, and font reproduces these as the start character § (Chr$(167)) and the stop character ¬ (Chr$(172)).

However, any alternative may be chosen from the list below - although not all Codabar readers recognise all these combinations.

Allowed start/stop codes: A B C D E N T *

All codes prepared to create a Codabar symbol must a start and stop character included.

Codabar does not use check digits, so there is no entry for Codabar in the dFont program.

Interleaved 2-of-5

Interleaved 2-of-5 (I 2/5) is a high density continuous symbology which encodes numeric digit pairs only. As this covers the sequence 00 - 99, clearly all digits pairs must be translated into single characters for this symbology to be represented by a font.

It should be noted that I 2/5 symbols require an even number of digits. The convention is that if an odd number of digits is to be encoded a LEADING 0 is attached to the code. This will of course decode when the symbol is subsequently scanned.

This code must begin with a start character and must end with a stop character. The start character is § (Chr$(167)) and the stop character is ¬ (Chr$(172)).

For convenience in typing the open curly bracket { may be used as the start character, and the close curly bracket } may be used as the stop character.

Digit pairs are represented in the dLSoft fonts by characters with an ASCII value corresponding to value of the digit pair. Thus the digit pair 65 is represented by A, because this has an ASCII value of 65.

ASCII values below 32 are not represented by readable characters and so for convenience of handling characters with ASCII values in the range 0 - 32 are reproduced in the font at the character positions chr$(192) - chr$(223).

I 2/5 are often used with a Modulo 10 check digit in the final position.

The dFont program provided with the dLSoft Barcode fonts calculates the check digit for I 2/5 codes, and the source code is provided and may be adapted to the users needs - for example, for automating the calculation in a spreadsheet or database.

Unfortunately I 2/5 suffer from the fact that a partial scan is likely to decode as a valid (although shorter) code, and the presence of a check digit does little to overcome this. However, some scanners may be set to accept a fixed number of digits and record an error if less than that number decode.

Interleaved 2-of-5 is the same encoding scheme and bar pattern as the ITF and ITF-6 outer case markers used in distribution, although the latter codes are normally printed with large bearer bars (although this is to spread the pressure during printing and has no effect on the scanning).

MSI/Plessey

MSI, also know as Modified Plessey Code, is a relatively weak code which is inefficient in its use of space.

This code must begin with a start character and must end with a stop character. The start character is § (Chr$(167)) and the stop character is ¬ (Chr$(172)).

For convenience in typing the open bracket (may be used as the start character, and the close bracket) may be used as the stop character.

Normally this code has a single Modulo 10 check digit, somewhat peculiar algorithm used for this is include in the dFont program. However, there are two variations of a double check digit form in common use. One uses a (standard) Modulo 11 check digit before the Modulo 10 check digit, the other uses two Modulo 10 check digits.

Some scanning equipment cannot read both forms. (in fact some scanning equipment cannot read either of the two check digit forms). Check your scanners documentation to ensure that you choose an appropriate combination.

The dFont program provided with the dLSoft Barcode fonts calculates the single Modulo 10 check digit for MSI codes, and the source code is provided and may be adapted to the users needs - for example, for automating the calculation in a spreadsheet or database.

Royal Mail RM4SCC

The RM4SCC is a clocked code which uses a central track made up of evenly spaced small bars, with the data encoded by the bars extending above and/or below the clocking bars.

The character set for RM4SCC consists of upper case letters and numbers only. The dLSoft barcode font includes a zero width space character, so that text copied from conventionally spaced postcodes may be translated into a barcode symbol.

This code must begin with a start character and must end with a stop character. The start character is § (Chr$(167)) and the stop character is ¬ (Chr$(172)).

For convenience in typing the open bracket (may be used as the start character, and the close bracket) may be used as the stop character.

The check digit is normally obtained by a table-lookup procedure, but a simple algorithm may also be used. The character values are shown in the table.

Character	**Value**
0 - 9	0 - 9
A - Z	10 - 35

The character values may be determined directly from the corresponding ASCII values, and then two parameters calculated (tu and tl) representing the row and column of the 6*6 table-lookup.

```
For i = 1 to len(code$)
      z=ASC(MID$(code$,I,1))
      If (z < 65) Then
            z = z - 48
         Else
            z = z - 55
          End If

         ii = Int(z / 6)
         If (ii >= 5) Then k = 0 Else k = ii + 1
         tu = tu + k' row ref
         ii = Int(z - ii * 6)
         If (ii >= 5) Then k = 0 Else k = ii + 1
         tl = tl + k' col ref

   Next i
```

Then the check digit value is calculated

```
   tu = tu Mod 6: If (tu = 0) Then tu = 6      ' checksum
   tl = tl Mod 6: If (tl = 0) Then tl = 6
   k = (tu - 1) * 6 + tl - 1
```

And finally the value is converted to ASCII

```
   If (k < 10) Then
      chn = (k + 48)
   Else
      chn = (k + 55)
```

```
checkchar$ = Chr$(chn)
```

The dFont program provided with the dLSoft Barcode fonts calculates check digits for RM4SCC codes, and the source code is provided and may be adapted to the users needs - for example, for automating the calculation in a spreadsheet or database.

Some other European countries use a virtually identical 4 state clocked code for barcoding the mail. In some of these countries the start and stop characters are NOT used, and in some case the check digit is not used. In such cases the text of the postcode may be reproduced directly using the dLSoft barcode font.

In all cases the size of the printed barcode is important, and a font size of 20 point should always be used.

PostNet

PostNet codes are the clocked codes used in the US mail system. There a three types of PostNet code (identified as A, C and C') which differ in the number of characters encoded. These codes are based on the US ZIP code system and use numbers only.

Code type	number of code characters
A	5
C	9
C'	11

This code must begin with a start character and must end with a stop character. The start character is § (Chr$(167)) and the stop character is ¬ (Chr$(172)).

For convenience in typing the open bracket (may be used as the start character, and the close bracket) may be used as the stop character.

For convenience of use the dLSoft barcode font also allows the (to used as a start character, and the) to be used for the stop character.

PostNet codes require a single check digit which may be calculated by adding up the numerical value of each character in the code and setting the check digit equal to the character which represents the number which must be added to the sum to give a multiple of 10.

The dFont program provided with the dLSoft Barcode fonts calculates check digits for PostNet codes, and the source code is provided and may be adapted to the users needs - for example, for automating the calculation in a spreadsheet or database.

Code 128/ EAN-128

Code 128 and EAN-128 are modern very high density coding schemes which are somewhat more complex than most other schemes. They have three coding schemes each and permit the inclusion of special characters not present on the keyboard. In general, because of the presence of non-printing characters, 128 type codes are best reproduced using the fonts which do NOT include text under the bars, such human readable text being added separately.

There are 105 characters in the character set, but each one may be used to represent more than one human readable character - see the table below. For example, the character with a Code 128 value of 77 represents a carriage return (CHR$(13)) in coding scheme A, the lower case letter m in scheme B, and the digit pair 77 in scheme C. This may be created in Basic using CHR$(77+32) = CHR$(109) [the coding scheme starts at ASCII 32].

Generally scheme B is used by default. For EAN-128 scheme C is used for any code which has numbers in the first four digits (as recommended by the ANA). The scheme must be specified by making the first character one of the start characters specified below. The stop character is always the same and is provided as ¬ (CHR$(172)) in the dLSoft barcode font.

In an attempt to simplifying the task of encoding characters the dLSoft 128 font provides the entire ASCII character set; obviously some of the 105 Code 128 characters are duplicated. ASCII values from 0 to 31 are available within the font as duplicates of the ASCII values 192 to 223. So setting CHR$(27) into the barcode font will produce the correct pattern, as will setting the character Û. The character Û has an ASCII value of 219 = 192 + 27.

Some of the control characters (values above 127) are not defined in some text fonts - so tend to show up as identical blocks. To make strings containing these characters easier to read before they become barcode symbols, the characters from CHR$ 128 to 137 are duplicated at 161 to 170. The latter are defined, as you can verify by holding down the ALT key and typing 0161 in Notepad.

The 128 coding standard allows the scheme to be changed within the symbol. Thus a symbol which starts with scheme C may be changed to scheme B part way through by using one of the special function codes (Code B). It is beyond the scope of this document to provide a tutorial in the use of 128 codes, and the user who needs more information should contact his national Article Numbering Association for details of the schemes and practices adopted locally.

The Code 128 character set is reproduced in the table below.

128 codes have a mandatory check digit which is calculated using a Modulo 103 of the weighted sum of the other characters in the code, where the weightings are determined by the character position (starting from the left). The result must have 32 added to it to enable the check digit character to be inserted as CHR(x).

The dFont program provided with the dLSoft Barcode fonts calculates check digits for Code 128 codes, and the source code is provided and may be adapted to the users needs - for example, for automating the calculation in a spreadsheet or database.

More:

EAN-128

EAN-128

EAN-128 is based on the Code 128 symbology, and scanning requirements are identical. The EAN-128 code is distinguished from Code 128 by having a Function 1 character immediately after the start character.

The value of the Function 1 character IS taken into account in creating the check digit.

The dFont program provided with the dLSoft Barcode fonts calculates check digits for EAN-128 codes, and the source code is provided and may be adapted to the users needs - for example, for automating the calculation in a spreadsheet or database.

More:

Code 128 character code, ASCII Table and font values

Code 128 character code, ASCII Table and font values

CodeA	CodeB	CodeC	Value	ASCII additional
space	space	00	0	32
!	!	01	1	33
"	"	02	2	34
#	#	03	3	35
$	$	04	4	36
%	%	05	5	37
&	&	06	6	38
'	'	07	7	39
(	(	08	8	40
)	)	09	9	41
*	*	10	10	42
+	+	11	11	43
,	,	12	12	44
-	-	13	13	45
.	.	14	14	46
/	/	15	15	47
0	0	16	16	48
1	1	17	17	49
2	2	18	18	50
3	3	19	19	51
4	4	20	20	52
5	5	21	21	53
6	6	22	22	54
7	7	23	23	55
8	8	24	24	56
9	9	25	25	57
:	:	26	26	58
;	;	27	27	59
<	<	28	28	60
=	=	29	29	61
>	>	30	30	62
?	?	31	31	63
@	@	32	32	64
A	A	33	33	65
B	B	34	34	66
C	C	35	35	67
…..	all uppercase letters			
X	X	56	56	88
Y	Y	57	57	89
Z	Z	58	58	90
[	[	59	59	91
\	\	60	60	92
]	]	61	61	93
^	^	62	62	94

_	_	63	63	95	
NUL	`	64	64	0	192
SOH	a	65	65	1	193
STX	b	66	66	2	194
ETX	c	67	67	3	195
EOT	d	68	68	4	196
END	e	69	69	5	197
ACK	f	70	70	6	198
BEL	g	71	71	7	199
BS	h	72	72	8	200
HT	i	73	73	9	201
LF	j	74	74	0	202
VT	k	75	75	11	203
FF	l	76	76	12	204
CR	m	77	77	13	205
SO	n	78	78	14	206
SI	o	79	79	15	207
DLE	p	80	80	16	208
DC1	q	81	81	17	209
DC2	r	82	82	18	210
DC3	s	83	83	19	211
DC4	t	84	84	20	212
NAK	u	85	85	21	213
SYN	v	86	86	22	214
ETB	w	87	87	23	215
CAN	x	88	88	24	216
EM	y	89	89	25	217
SUB	z	90	90	26	218
ESC	{	91	91	27	219
FS	|	92	92	28	220
GS	}	93	93	29	221
RS	~	94	94	30	222
US	DEL	95	95	31	223
Func3	Func3	96	96	128	161
Func2	Func2	97	97	129	162
Shift	Shift	98	98	130	163
Code C	Code C	99	99	131	164
Code B	Func4	CodeB	100	132	165
Func4	Code A	Code A	101	133	166
Func1	Func1	Func1	102	134	167
StartA	StartA	StartA	103	135	168
StartB	StartB	StartB	104	136	169
StartC	StartC	StartC	105	137	170
Stop	Stop	Stop	106	138	172

Telepen

The Telepen coding scheme has a number of variants - Telepen Numeric, Telepen ASCII, and either with various begin and end codes. You will need to determine which variant you are attempting to create barcodes for.

The Telepen ASCII scheme provides the full ASCII character set. For convenience of handling characters with ASCII values of less than 32 are reproduced in the font at the character positions chr$(192) - chr$(223).

Some Telepen symbols require the first character (after start) be an ASCII Shift In character, and the last character (before stop) to be a Shift Out character. The ASCII ESC character is required on some Telepen Numeric systems as the first character after to the start character. If you are using a Telepen system your system manual describes the scheme you require.

All Telepen codes must begin with a start character and must end with a stop character. The start character is § (Chr$(167)) and the stop character is ¬ (Chr$(172)).

Telepen ASCII normally uses a Modulo 127 check digit as shown below. However, there seem to be a wide variety of schemes for calculating Telepen check digits and the user is recommended to consult his Telepen system documentation.

```
n = Len(xx$)   ' xx$ is the code
t1 = 0
For i = 1 To n
    y$ = Mid$(xx$, i, 1)
        z = Asc(y$) ' ASCII value of character
        If (z > 127) Then z = z - 128
        t1 = t1 + z
        Next i
    chn = t1 Mod 127   ' checksum
    If chn > 0 Then chn = 127 - chn
    If (chn < 32) Then chn = chn + 192 ' make it visible in font
    chd$ = Chr$(chn)    ' set check character
```

The dFont program provided with the dLSoft Barcode fonts check digit for Telepen codes, and the source code is provided and may be adapted to the users needs - for example, for automating the calculation in a spreadsheet or database.